



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**DESIGN AND LABORATORY IMPLEMENTATION OF
AUTONOMOUS OPTIMAL MOTION PLANNING FOR
NON-HOLONOMIC PLANETARY ROVERS**

by

Travis K. Bateman

December 2012

Thesis Co-Advisors:

Mark Karpenko
I. Michael Ross

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 2012	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE DESIGN AND LABORATORY IMPLEMENTATION OF AUTONOMOUS OPTIMAL MOTION PLANNING FOR NON-HOLONOMIC PLANETARY ROVERS			5. FUNDING NUMBERS	
6. AUTHOR(S) Travis K. Bateman				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) This work investigates the challenge of designing and implementing minimum-time trajectories for an autonomous, non-holonomic, planetary rover. The optimal trajectories were implemented at the Control and Optimization Laboratories with a TRAXXAS remote controlled vehicle modified to enable autonomous operations. These modifications include the addition of an ArduPilot controller into the architecture of the vehicle. The ArduPilot controls the inputs to the drive motor and steering servos to implement the trajectory commands generated by the trajectory optimization tool, DIDO. The challenging problem of parallel parking was used to evaluate a canonical maneuvering scenario and illustrate a procedure for motion planning that could be used for guiding a planetary rover. Three cases were evaluated with different starting points to illustrate the difficulties associated with controlling a non-holonomic vehicle. The starting points were located in front of, next to, and behind the parking space. In addition to each case, three scenarios were evaluated for complexity: no cars, two cars parked with an ideal amount of space between them, and two cars parked with minimal space between them. A VICON motion capture system was used measure the vehicle trajectory in experiments.				
14. SUBJECT TERMS Pseudospectral Optimal Control, Trajectory Optimization, Path Planning, Robotics Manipulator, Collision Avoidance, DIDO, Autonomous Robotic Control			15. NUMBER OF PAGES 93	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**DESIGN AND LABORATORY IMPLEMENTATION OF AUTONOMOUS
OPTIMAL MOTION PLANNING FOR NON-HOLONOMIC PLANETARY
ROVERS**

Travis K. Bateman
Lieutenant, United States Navy
B.S., University of San Diego State, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
December 2012**

Author: Travis K. Bateman

Approved by: Mark Karpenko
Thesis Co-Advisor

I. Michael Ross
Thesis Co-Advisor

Knox T. Millsaps
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This work investigates the challenge of designing and implementing minimum-time trajectories for an autonomous, non-holonomic, planetary rover. The optimal trajectories were implemented at the Control and Optimization Laboratories with a TRAXXAS remote controlled vehicle modified to enable autonomous operations. These modifications include the addition of an ArduPilot controller into the architecture of the vehicle. The ArduPilot controls the inputs to the drive motor and steering servos to implement the trajectory commands generated by the trajectory optimization tool, DIDO.

The challenging problem of parallel parking was used to evaluate a canonical maneuvering scenario and illustrate a procedure for motion planning that could be used for guiding a planetary rover. Three cases were evaluated with different starting points to illustrate the difficulties associated with controlling a non-holonomic vehicle. The starting points were located in front of, next to, and behind the parking space. In addition to each case, three scenarios were evaluated for complexity: no cars, two cars parked with an ideal amount of space between them, and two cars parked with minimal space between them. A VICON motion capture system was used measure the vehicle trajectory in experiments.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	MOTIVATION	1
	1. Current Mars Lander.....	1
	2. Past Planetary Rovers.....	2
	<i>a. Twin Rovers - Spirit and Opportunity</i>	<i>2</i>
	<i>b. Navigation/Mobility</i>	<i>3</i>
	3. Future Exploration	3
	4. Routine Operations.....	4
B.	THESIS OUTLINE.....	4
II.	EXPERIMENTAL SETUP	7
A.	OVERVIEW OF EXPERIMENTAL SETUP.....	7
	1. Traxxas-Summit Vehicle	8
	2. Modifications to the Traxxas Vehicle.....	9
	<i>a. Rewiring the Input Channels</i>	<i>10</i>
	<i>b. MUX/failsafe</i>	<i>11</i>
	<i>c. Stiffening the Suspension</i>	<i>11</i>
	<i>d. New Motor Controller.....</i>	<i>12</i>
	3. ArduPilot Mega.....	13
	<i>a. Hardware Description.....</i>	<i>14</i>
	<i>b. MUX</i>	<i>14</i>
	<i>c. Powering Up the Vehicle</i>	<i>16</i>
B.	VICON MEASUREMENT SYSTEM	16
	1. Hardware.....	16
	2. Software	19
III.	VEHICLE MODELING AND CALIBRATION	21
A.	VEHICLE KINEMATIC MODEL	21
	1. The Four-Wheeled Car.....	21
	2. The Need for Model Calibration	22
B.	MODEL CALIBRATION.....	22
	1. Turning Angle (γ) and Steering	22
	2. Vehicle Speed Calibration.....	25
	3. Testing of Calibration.....	27
IV.	A CANONICAL MOTION PLANNING PROBLEM	31
A.	PARALLEL PARKING SCENARIO.....	31
	1. Parallel Parking	31
	2. Parallel Parking in Minimum-time	34
	3. Validating the Minimum-Time Problem	35
	4. Using the DIDO Tool	37
	5. Modeling Obstacles.....	37
B.	TRAJECTORY VALIDATION	39
	1. Hamiltonian	39

2.	Costates	40
3.	Feasibility Test	40
4.	The Path Function Affects the Hamiltonian and the Costates	40
V.	EXPERIMENTAL RESULTS.....	43
A.	INTRODUCTIONS	43
B.	TIME-OPTIMAL SOLUTIONS	44
C.	IMPLEMENTATION	48
VI.	CONCLUSIONS AND RECOMMENDATIONS.....	53
A.	CONCLUSION	53
B.	RECOMMENDATION FOR FUTURE WORK.....	53
	APPENDIX.....	55
	LIST OF REFERENCES.....	75
	INITIAL DISTRIBUTION LIST	77

LIST OF FIGURES

Figure 1.	Mars Rover “Curiosity” from [3].	2
Figure 2.	Mars Exploration Rover “Spirit” or “Opportunity from [3].	3
Figure 3.	Schematic of Mother Ship Concept Supporting Multiple Planetary Rovers.	4
Figure 4.	Overall Schematic of the Autonomous Unmanned Vehicle Setup.	7
Figure 5.	Dimensions of the Traxxas-Summit, from [7].	8
Figure 6.	7-cell NiMH 8.4V Battery Pack, from [7].	8
Figure 7.	Clear Platform With All Components.	9
Figure 8.	The Communication Schematic for the Vehicle	10
Figure 9.	TRAXXAS TQ 2.4 GHz Radio Remote Controller, from [7].	11
Figure 10.	Stiffening the Suspension: (a) Original Springs, from [7], and (b) Rigid Aluminum Cylinders.	12
Figure 11.	Modification to the Vehicle Speed Controller: (a) Original EVX-2, from [7]; (b) New HB-25 Motor Controller, from [10].	13
Figure 12.	ArduPilot Mega Autopilot Board, from [8].	13
Figure 13.	Three LEDs for Power, MUX, and the PPM, from [8].	15
Figure 14.	AVRISP MKII programmer attached to the ArduPilot Mega board to flash the MUX, from [8].	15
Figure 15.	VICON Motion Capture System (a) Physical Markers, (b) IR Camera, and (c) Giganet Interface from [11].	17
Figure 16.	Screen Shot of the VICON Tracker Software.	17
Figure 17.	VICON Stacked Above One Another in the Laboratory.	18
Figure 18.	(a) Side View and (b) Top View of the Configuration of the Six-Cameras in the VICON Tracker Program.	19
Figure 19.	Traxxas Vehicle Roof with Three Markers in an L-Shape.	20
Figure 20.	Tracker Program with the Traxxas Vehicle Model.	20
Figure 21.	Model of a Front-Wheel Steering Vehicle.	21
Figure 22.	Evaluating the γ Angle.	23
Figure 23.	Visual Depiction of Determining Length of D.	24
Figure 24.	Experimental Calibration Curve for Steering Servos.	25
Figure 25.	Experimental Velocity Profiles with Linear Trends.	26
Figure 26.	Velocity Profile Using Equation (6) and (7).	27
Figure 27.	Velocity Profile After Tuning the Velocity Slope Using Equation (8) and (9).	28
Figure 28.	Steering Calibration Test for Controlling the Vehicle Orientation.	29
Figure 29.	Schematic of the Geometry of Perfect Parallel Parking after [13].	33
Figure 30.	Schematic of Lengths for a Geometrically Perfect Parallel Parking.	33
Figure 31.	Vehicle Represented in DIDO With Two Circles.	37
Figure 32.	Model of Vehicle, Obstacle, and Barrier	38
Figure 33.	Minimum Distance Between Two Circles.	39
Figure 34.	Example of the Hamiltonian With a Constant Value of Negative One During the Entire Commanded Trajectory (Problem given in 13 to 16).	39
Figure 35.	Graph of Costates.	40

Figure 36.	Comparison of the Optimal Solution to the Propagated Solution from ODE 45 (Problem in 13 to 16).	41
Figure 37.	The Effect of the Path Function on Costates.	42
Figure 38.	Schematic of the Initial Positions of the Three Parallel Parking Cases.	43
Figure 39.	Optimal Trajectories for Scenario With No Cars	45
Figure 40.	Optimal Trajectories for Scenario With Ideal Spacing	46
Figure 41.	Optimal Trajectories for Scenario With Minimal Spacing	47
Figure 42.	Experimental Results of All Three Cases for the Ideal Parking Space.....	48
Figure 43.	Experimental Results of All Three Cases for the Minimal Parking Space.....	49
Figure 44.	Still Photos of the Experiment for Scenario Three: Case 1.	50
Figure 45.	Still Photos of the Experiment for Scenario Three: Case 2.	51
Figure 46.	Still Photos of the Experiment for Scenario Three: Case 3.	52

LIST OF TABLES

Table 1.	Optimal Times to Parallel Park.....	47
Table 2.	Experimental Time for Parallel Parking.	49

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Professor Michael Ross, I must say thank you for entrusting me with this chance to apply the knowledge that I have gained while here at Naval Postgraduate School. It has been an honor to be able to apply the theory of optimal control to an actual vehicle.

Professor Mark Karpenko, I could not have done this without you. Thank you for the many hours you dedicated to assisting with my thesis. I respect your vision, guidance, and insight with the use of robotics and computers. Thank for sharing your suggestions, comments and concerns. You have been a great example of examining experimental data and finding the positive from any results.

Robert Casey and Brad Watanabe, thank you for your pool of knowledge that I always tapped into when I was stuck in my thesis process. You made my life better with your willingness to share your experience. Thank you for your time when I needed a sounding board when things were just not going my way. You two are the best.

Above all, I am thankful for a Heavenly Father that loves me and His Son Jesus Christ who died for me. I am honored that My Heavenly Father has entrusted me with His precious daughter of God, Kate, to be my eternal companion. Thank you my Love for supporting me in all that I do and will do. I love you!

To Megan, Austin, and Morgan; thank you for your warm hugs everyday as you greet me when I arrive home. I hope that you always strive for new opportunities to learn and to never take education for granted. I love you!

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

1. Current Mars Lander

The recent terrestrial rover Curiosity in Figure 1 successfully landed on the surface of Mars on the 6th of August 2012. Curiosity explored the terrain of Mars semi-autonomously as it conducted its experiments along its daily pre-planned path to its next site. The maneuvering of the Martian rover over the dusty and rocky terrain of Mars was controlled by a combination of human-in-the-loop intervention and pre-programmed trajectories for a semi-autonomous function. This combination of human control and autonomy helped to maximize a Mars day journey, because humans could only plan the first portion of the journey based on images sent from the rover. After the pre-planned trajectory was concluded the autonomy function would take over to allow the rover to continue as long as it could determine a safe trajectory based on its sensors [1].

The distance between Mars and Earth ranges from 100-200 million miles [1] and at this distance, the time delay in the communication makes it impossible to directly control planetary rovers. A human could not drive Curiosity in real-time by utilizing video feed from a camera mounted on the Mars rover and then steering the vehicle from the time delayed video once the images reached earth. In the case of Curiosity, it could be stuck in a sand trap or even worse run off a cliff before the video would display the hazard. Human drivers for the Mars rovers control the vehicle through trajectory commands. These commands are up-loaded to the vehicle daily from images obtained the previous day. During the autonomous function, the rover relies on sensory feedback to make decisions on the pre-planned trajectory for the vehicle such as avoiding hazards. Curiosity, to date, has little published literature on the details of its surface navigation and mobility control because it is still so new. The Curiosity control technology did stem from its predecessor twin rovers *Spirit* and *Opportunity*. An obvious technology upgrade for Curiosity was the power source, as it was nuclear vice solar [2]. This expanded the mobility range of the rover and enabled it to travel day or night.

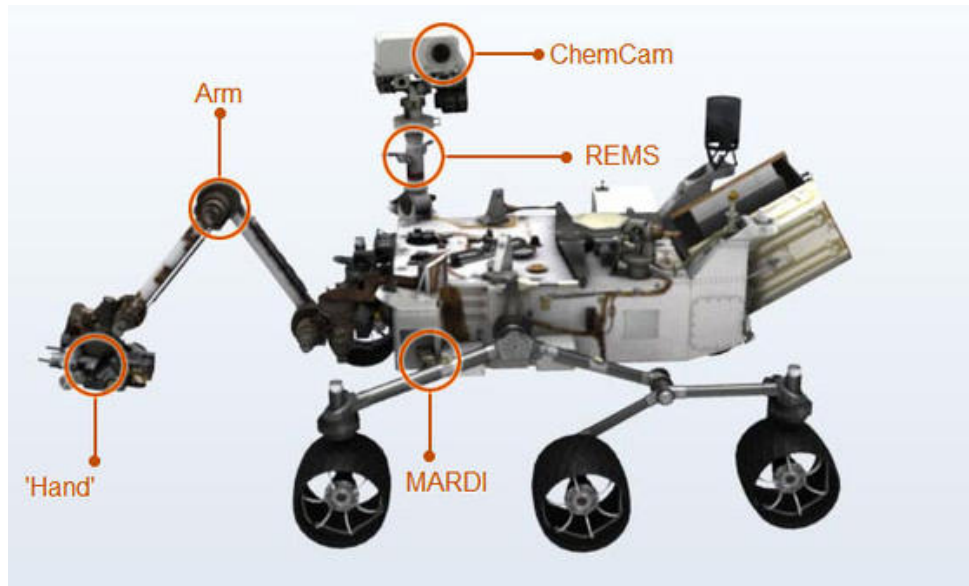


Figure 1. Mars Rover “Curiosity” from [3].

Curiosity performed many routine maneuvers and scientific experiments through its life on Mars. The more ground Curiosity covers and is able to explore will allow for further and expanded explorations of our neighboring red planet in the future. This additional knowledge of Mar’s surface will make it possible for rovers in the future to be fully autonomous in routine operations.

2. Past Planetary Rovers

a. *Twin Rovers - Spirit and Opportunity*

NASA’s program Mars Exploration Rovers (MER) launched two twin rovers to the surface of Mars in January 2004. They were deployed into two different locations on the surface of Mars. The primary mission consisted of 90 Mars solar days. The MER rovers lasted for three years. The rovers had mobility and vision mounted technologies that allowed for many semi-autonomous functions. Despite this technology, the rovers only traveled an average velocity of 0.2 miles/hour, or 38 yards/hour while in autonomous mode [4].



Figure 2. Mars Exploration Rover “Spirit” or “Opportunity” from [3].

b. Navigation/Mobility

Vital parts of autonomous mobility are the ability to perform onboard motion planning as well as to detect obstacles and hazards and avoid them. MER vehicles had the ability to perform terrain assessment, and used an autonomous driving mode called Local Path Selection. The Local Path Selection had the rovers conduct path corrections when traveling to a pre-determined location. This feature worked for the pre-planned maneuvers, which were planned by humans the day before based on available sensory and visual inputs. Once the vehicle had completed the pre-planned maneuvers, the rover continued autonomously into unknown territory [1].

3. Future Exploration

Mars could become a place of increased exploration through the use of multiple planetary land rovers. Multiple rovers could require a mother ship to support communications and recharging of the individual land rovers. In this scenario (see Figure 3 for a artists’ rendition), the land rovers conduct routine operations, specifically the maneuvering of the vehicle into a designated position for docking with the mother ship.

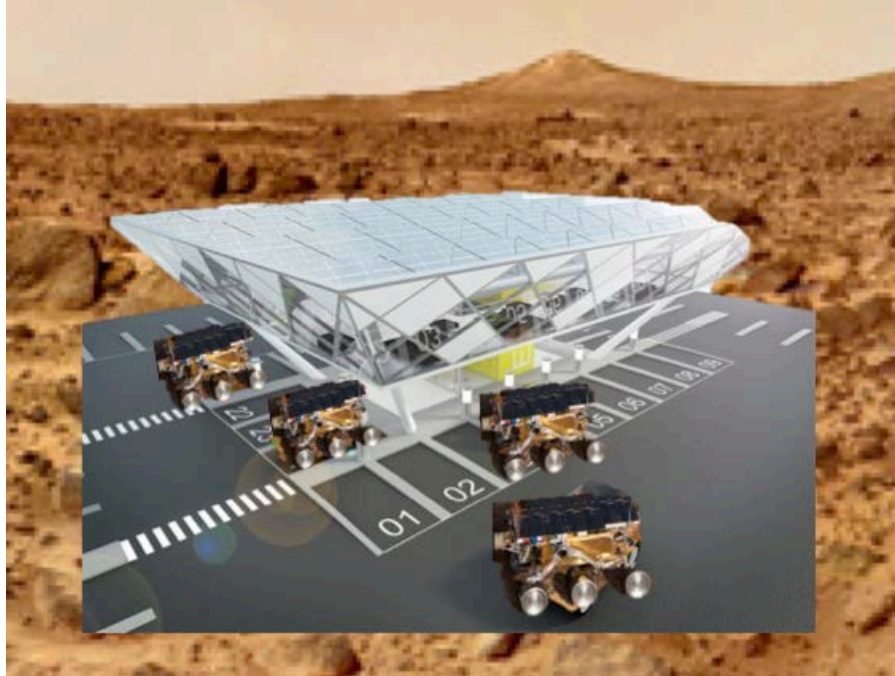


Figure 3. Schematic of Mother Ship Concept Supporting Multiple Planetary Rovers.

4. Routine Operations

Every day, routine operations are conducted on Earth with non-holonomic vehicles. One such operation is parking. Specifically, a challenging scenario for a non-holonomic would be the parallel parking scenario. What makes motion planning challenging for non-holonomic vehicles is the steering of the wheels. The steering wheels can be angled with respect to the alignment of the vehicle and move independently from the vehicle's orientation. But, the vehicle cannot be moved arbitrarily. The combination of both the velocity of the vehicle and the steering angle over time creates a specific trajectory. The combination of several path segments in different directions can produce a specific trajectory to obtain a desired orientation of the vehicle. Therefore, to move a vehicle into a parking space, a vehicle not only needs a path from point A point B, but also a specific set of velocity and steering trajectories.

B. THESIS OUTLINE

Conducting autonomous missions on the surface of Mars is challenging. Similar challenges are applied to operating autonomous vehicles on Earth that may be only a few

hundred miles away. The excessive amount of time spent by humans preparing trajectories for so-called autonomous vehicles makes the cost of the missions high. To minimize cost and time required for trajectory planning, and also maximize the mission effectiveness, a tool to assist in trajectory planning is required. Pseudospectral motion planner is one approach for accomplishing this goal. This approach can find feasible solutions to motion/path planning problems while simultaneously maximizing a mission relevant objective function such as time, distance, or fuel [5].

This thesis focuses on the development of time-optimal trajectories for a rover vehicle utilizing the software tool DIDO [6]. Maneuvering trajectories were analyzed and implemented on a non-holonomic land vehicle to demonstrate the application of the approach. The scope of this thesis is limited to examining of parallel parking as a means to illustrate a procedure for motion planning and autonomous guidance that could be applied to a planetary rover.

The remainder of the thesis is outlined as follows. Chapter II discusses the experimental set up and modifications to the vehicle. The techniques and details of modeling and calibrating the vehicle are described in chapter III. Chapter IV details the scenario for a canonical motion planning, while chapter V presents the optimal solutions for a variety of parking scenarios and the outcomes of the experimental implementation. Chapter VI covers the conclusion and future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. EXPERIMENTAL SETUP

A. OVERVIEW OF EXPERIMENTAL SETUP

The overall architecture of the rover that was utilized in this thesis is found in Figure 4. The software program DIDO is used to produce an optimal trajectory solution for the robot. The DIDO program solves the trajectory in terms of velocity and steering commands. The details of how this is done will be described later in chapter III. The velocity and steering commands need to be converted to pulse width commands for the steering servos and motor controller on the robot. The pulse width commands are sent to the robot through a wired connection; however, a wireless system could be set up as well. The ArduPilot, a microcontroller located on the vehicle, then receives the commands and routes the command signals to the appropriate components in order to execute the trajectory. As the rover implements the trajectory, the movement is captured with a VICON motion capture system.

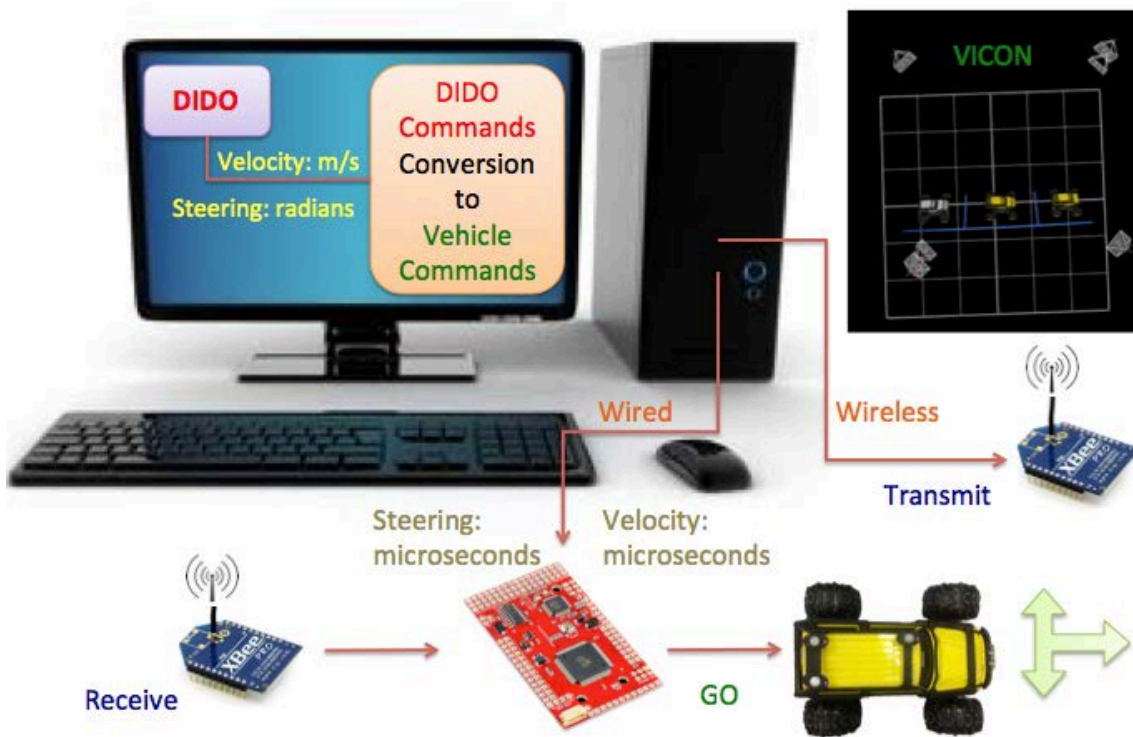


Figure 4. Overall Schematic of the Autonomous Unmanned Vehicle Setup.

1. Traxxas-Summit Vehicle

The rover used for this thesis was the Traxxas-Summit vehicle. The vehicle was a 1/10 scale Summit truck model 5607. See Figure 5 for dimensions of the vehicle. The remote controller was a TQ 2.4 GHz transmitter with five channels, controlling the throttle, steering, shifting to low and high gears, and the differential locking of the wheels. The vehicle contained a receiver with 5 output channels connected to four controlling servos. The remaining channel was used for the electronic speed control (ESP). Two 7-cell NiMH 8.4V stick packs (see Figure 6) supplied power to the vehicle.



Figure 5. Dimensions of the Traxxas-Summit, from [7].



Figure 6. 7-cell NiMH 8.4V Battery Pack, from [7].

2. Modifications to the Traxxas Vehicle

A clear platform (see Figure 7) was added to the car for the ease of mounting additional equipment to the vehicle. An ArduPilot, the autopilot board described in the next section, was added as an interface with the remote controller and the steering servos and motor controller on the vehicle. The communication path comes from either the remote controller or the computer (via a serial port) to the ArduPilot. From the ArduPilot, the commands are routed to the steering servos and motor controller as shown in Figure 8.

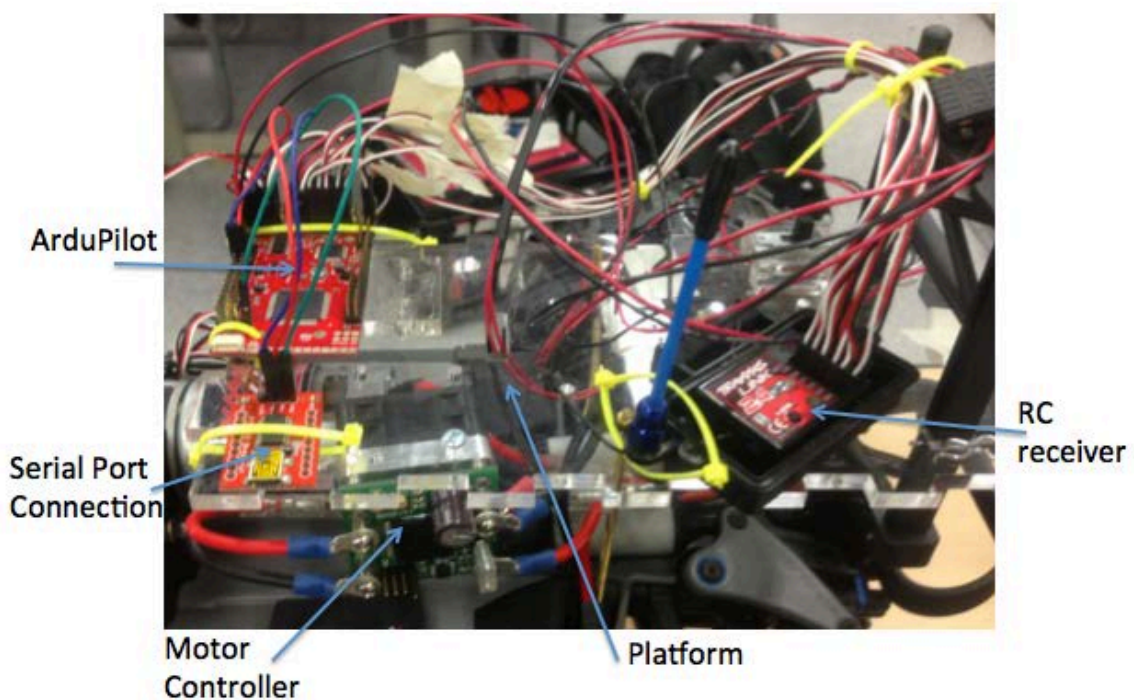


Figure 7. Clear Platform With All Components

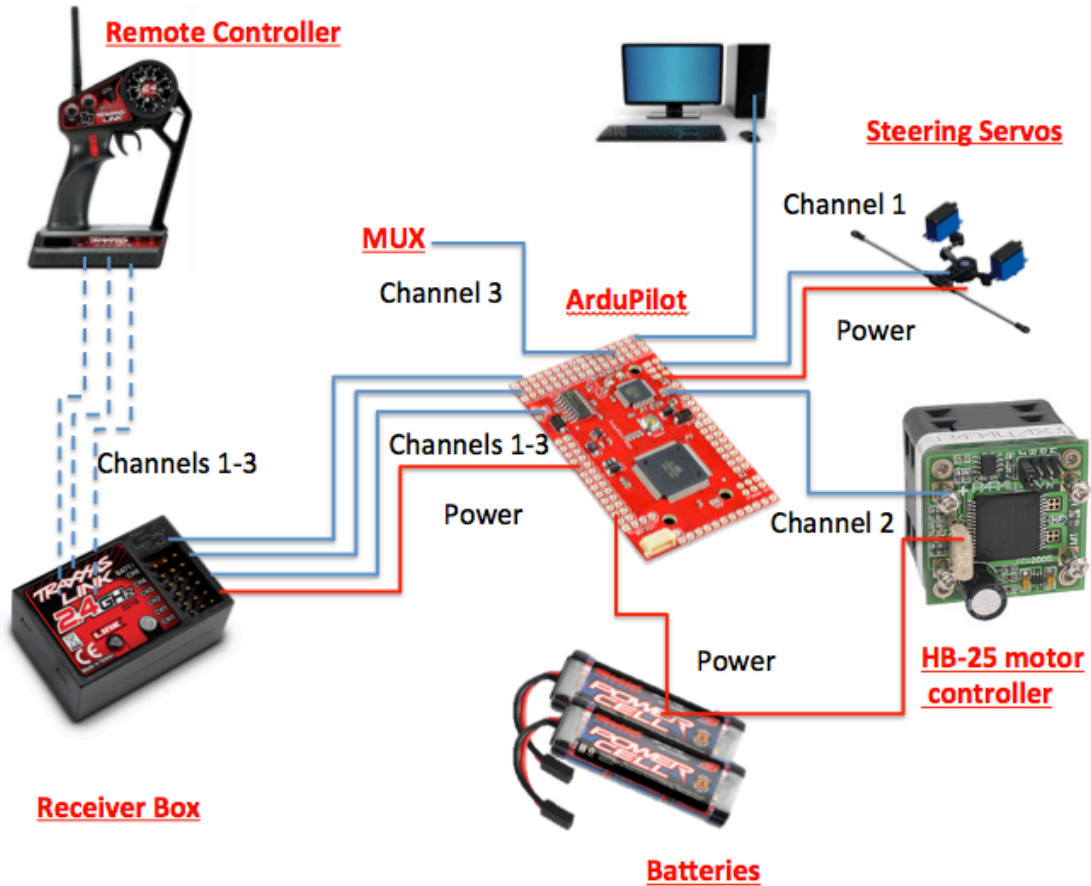


Figure 8. The Communication Schematic for the Vehicle

a. Rewiring the Input Channels

The Traxxas vehicle had been wired with five channels. The first channel controlled two steering servos; channel two controlled the EVX-2 Electronic Speed Control; channel three through five controlled the hi/low-shifting servo, and the front and rear differential lock (T-lock) servos. The shifting from hi to low gear was not needed (only low speeds would be used) and therefore the vehicle was set in low gear and then disconnected from the controller to ensure the vehicle remained in low gear. The front and rear T-locks, allowed for engagement to either lock or allow independent movement with left and right tires on each axle. The differential locks were disengaged to allow the wheels to move independently. This allowed for a smaller turning radius for the vehicle. The T-locks were then disconnected from the electronics to ensure they could not be accidentally engaged.

b. MUX/failsafe

The T-Lock switch located on the remote controller (see Figure 9) was rewired to control the ArduPilot MUX/failsafe. The position of the T-Lock switch controlled the power supply to the MUX. When power was supplied to the MUX, the yellow MUX light illuminated, and the ArduPilot/computer controlled the vehicle. Once power had been removed from the MUX, the yellow MUX light went out, and the remote controller manually controlled the vehicle.



Figure 9. TRAXXAS TQ 2.4 GHz Radio Remote Controller, from [7].

c. Stiffening the Suspension

The Traxxas vehicle suspension was built for racing and rugged terrain. The suspension allowed for additional movement of the vehicle and caused the vehicle to pitch and roll during quick starting and stopping. The additional movement was not problematic, but influenced the measurement of the position and velocity of the vehicle. Since the actual position and velocity could be used as a feedback source to close the control loop, it was desired to make the vehicle more rigid by replacing four suspension springs with aluminum cylinders having the same length and diameter as the springs (see Figure 10).



(a)



(b)

Figure 10. Stiffening the Suspension: (a) Original Springs, from [7], and (b) Rigid Aluminum Cylinders.

d. New Motor Controller

The Traxxas vehicle was not built for slow speeds. The EVX-2 electronic speed controller (see Figure 11) had a minimum speed that was too high for the laboratory. The Parallax HB-25 motor controller (see Figure 11) has been used in other vehicles in the lab, and was used to replace the EVX-2 on Traxxas vehicle. The HB-25 motor controller is a self-contained control system with an efficient thermal design and a high-current controller. It has a safety feature with a built-in automatic shut-off for invalid signals [10]. The upgraded motor allowed for much better low speed control of the Traxxas vehicle.

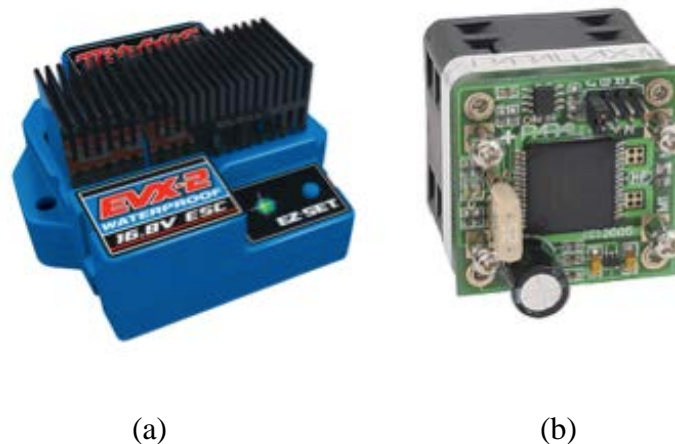


Figure 11. Modification to the Vehicle Speed Controller: (a) Original EVX-2, from [7]; (b) New HB-25 Motor Controller, from [10].

3. ArduPilot Mega

ArduPilot Mega, shown in Figure 12 is a microcontroller-based autopilot board and was added to the vehicle as an interface to enable a computer to command the vehicle. The benefit of the ArduPilot is that it is an off the shelf product, which is fully programmable to enable implementation of the optimal motion trajectories. This board adds the capability of switching from the autonomous/programmable mode to a manual/remote mode. The details of this action will be described later in this chapter.

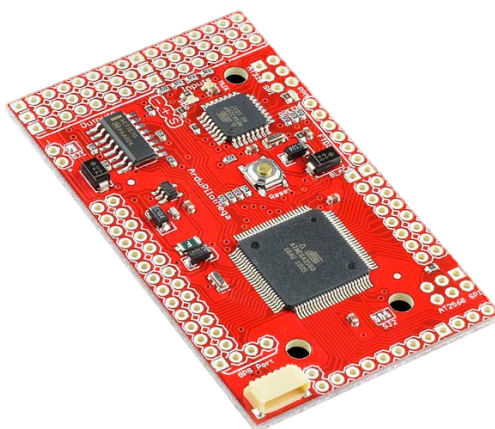


Figure 12. ArduPilot Mega Autopilot Board, from [8].

a. Hardware Description

The auto pilot board size is $1\frac{1}{2} \times 2\frac{5}{7}$ inches and is stable for ground, aerial, and watercraft vehicles. This board has a six-pin global position system (GPS) input and four-serial ports. The serial ports are used for wired or wireless communications from a computer or other sensors. Power to the board is supplied from one of two battery packs. The ArduPilot has LEDs that report the status of the board. The ATmega328 chip is the processor for the ArduPilot, which could be programmed by the user to implement any desired trajectory. The multiplexer (MUX) chip is used as a failsafe during the autonomous control of the vehicle. The failsafe function transfers control from the autonomous/programming mode to manual/remote mode and vice versa. The use of the MUX/failsafe assisted in two ways. First, it allowed manual shifting from the autopilot to the RC control mode during failures in the autopilot mode. This was especially advantageous when calibrating the vehicle. For example, if at any point the vehicle does not respond as expected, the failsafe switch is flipped and the vehicle motion stops. Second, the failsafe assists in trouble-shooting for the vehicle. For example, if the vehicle did not operate as desired, the switch allows for a quick check to make sure that the mechanical functions on the vehicle still works in the manual mode. More information about the ArduPilot can be found at [9].

b. MUX

The MUX/failsafe on the ArduPilot comes pre-programmed and ready for use in an autonomous system. The LEDs previously mentioned are shown in Figure 13. The power LED is red and lit when power is supplied to the board. The yellow MUX LED is lit when the MUX is activated in the autopilot/autonomous mode. The PPM LED has a different output when in the normal mode, the pass-through, or the failsafe mode as indicated in [8].

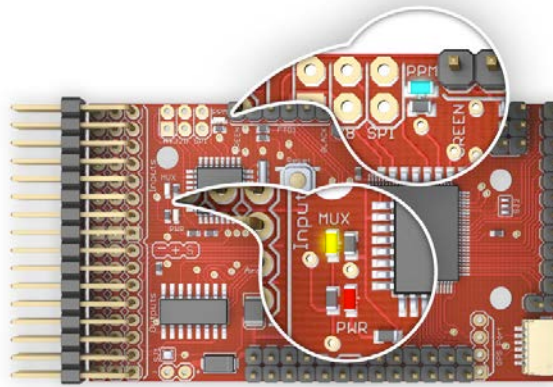


Figure 13. Three LEDs for Power, MUX, and the PPM, from [8].

During the first test of the ArduPilot board, the PPM's LED was lit, but it was not blinking. This lighting scheme for the PPM indicated that the MUX was not working properly. Flashing the MUX with up-to-date software was attempted to correct the problem with the MUX. To flash the MUX requires the AVRISP MKII programmer to be attached to the ArduPilot as shown in Figure 14. The AVR Studio and USB driver were installed on the programming computer to run the AVRISP MKII. Then the WINAVR program file was downloaded in preparation to program the MUX. The PPM encoder source codes were compiled and uploaded to the MUX. The location of files and additional information can be found in [8]. After flashing the MUX with the new firmware, the behavior of the blue PPM LED indicated the MUX was working properly.

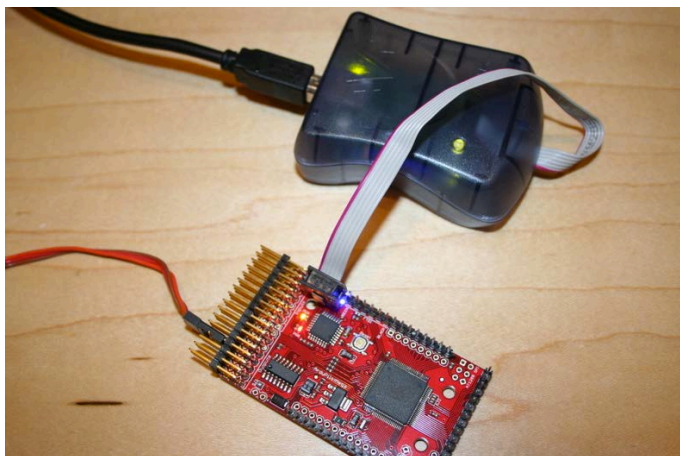


Figure 14. AVRISP MKII programmer attached to the ArduPilot Mega board to flash the MUX, from [8].

c. Powering Up the Vehicle

To prevent damage to the vehicle and its electronic components, a sequence must be followed when applying power to the vehicle. First, initiate power to the remote microcontroller and ensure the toggle switch is in the desired position (i.e. manual or ArduPilot/computer control). Next, plug the first battery pack into the adapter connected to ArduPilot and the motor. Finally attach the second battery pack to the other adapter connected to the motor only. Following the sequence of steps ensures that the microcontroller is running before power is applied to the HB-25 motor controller.

B. VICON MEASUREMENT SYSTEM

VICON is a motion caption system that was used to capture the position and calculate the velocity of the vehicle. The VICON system allows the comparison of the actual path of the vehicle to the optimal trajectory solutions solved later in this thesis.

1. Hardware

The VICON motion caption system is a tool to capture real-time motion through the use of multiple cameras. The camera captures a high-resolution image of a reflected marker at a specific wavelength. The lighting mounted around the camera lens, as seen in Figure 15 (b), produces the required specific wavelength. The markers, as shown in Figure 15 (a), are reflecting the specific wavelength back to the camera lens. The cameras then track the reflection as picture frames at a rate of 100 frames per second. These frames are then sent to the Giganet (seen in Figure 15 (c)). The Giganet interfaces with the computer and routes the data from the cameras to a program called VICON Tracker [11]. A screenshot of the Tracker software is shown in Figure 16.



Figure 15. VICON Motion Capture System (a) Physical Markers, (b) IR Camera, and (c) Giganet Interface from [11].

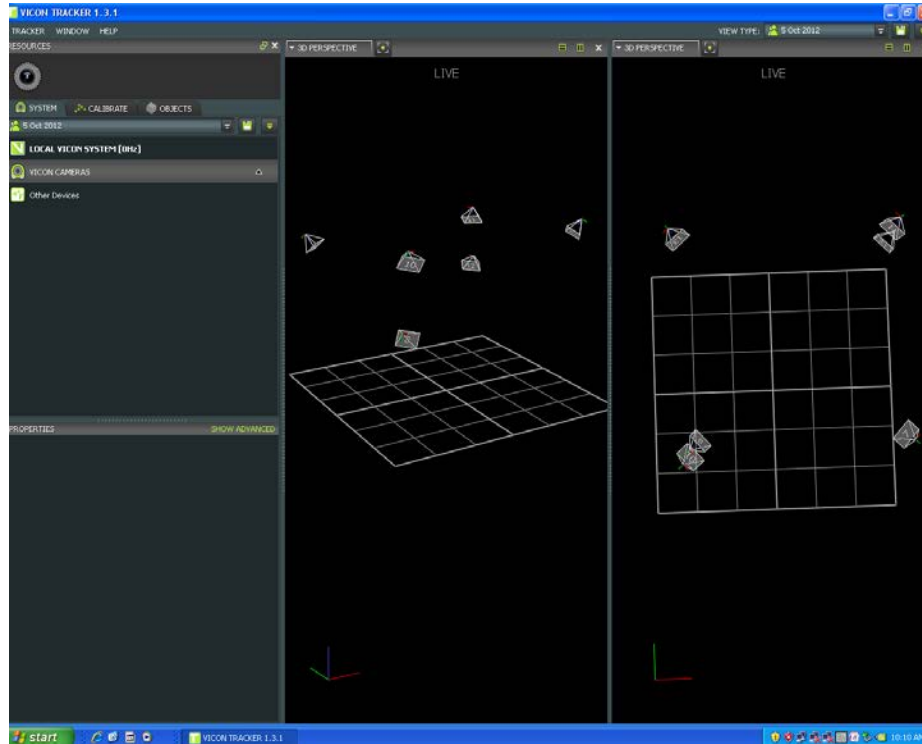
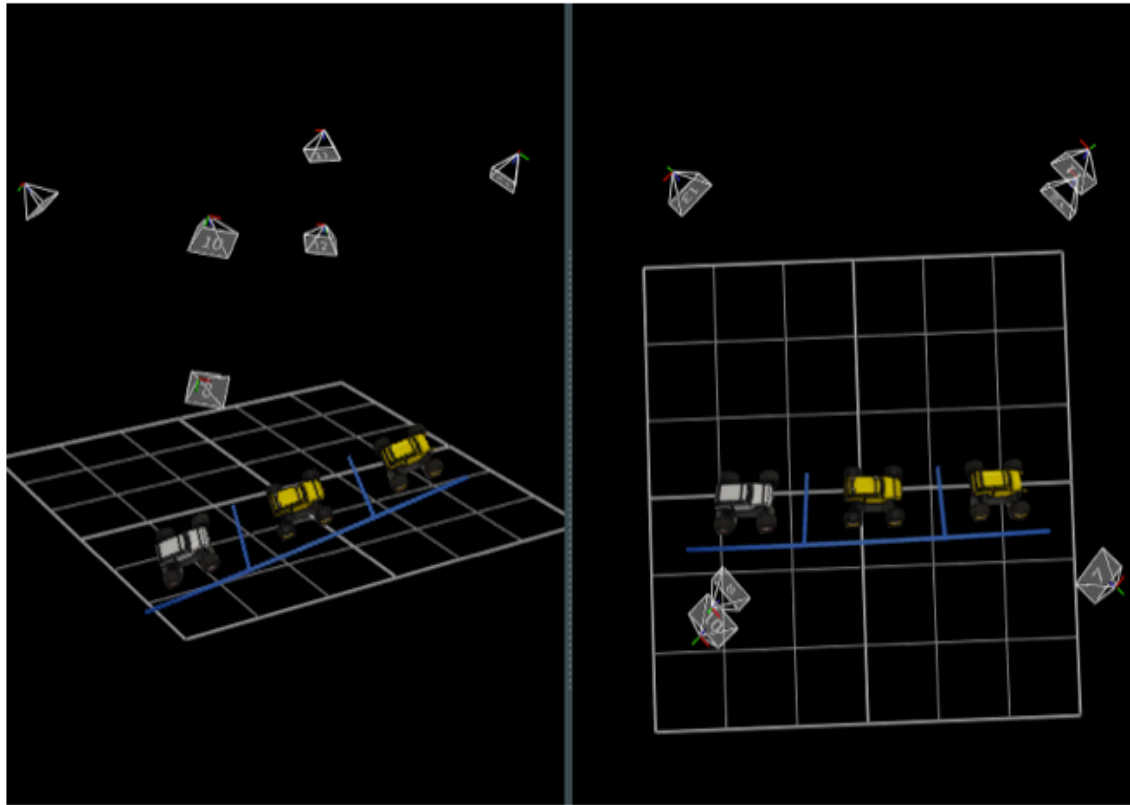


Figure 16. Screen Shot of the VICON Tracker Software.

The arrangement of cameras utilized a placement of six cameras, which were located at the corners of a square work area. Two of the corners had two cameras stacked, one above the other (see Figure 17). The other two corners had a single camera mounted approximately the same height in relation to the top camera of the two stacked camera configurations. Figure 18 is a screen shot from the program Tracker, which shows a representation of the six cameras and their orientation in the lab space. A parallel parking scenario is overlaid for reference.



Figure 17. VICON Stacked Above One Another in the Laboratory.



(a)

(b)

Figure 18. (a) Side View and (b) Top View of the Configuration of the Six-Cameras in the VICON Tracker Program.

2. Software

The VICON Tracker software was relatively easy to setup up. The program tracks the position of the Traxxas vehicle through the use of three reflective markers in an L-shape attached to the roof of the vehicle (see Figure 19). The markers were placed in an L-shape to allow the vehicle's orientation to easily be identified from the computer screen (see Figure20). The corner of the right triangle, created by the L-shape, correlated to the rear driver side of the vehicle.



Figure 19. Traxxas Vehicle Roof with Three Markers in an L-Shape.

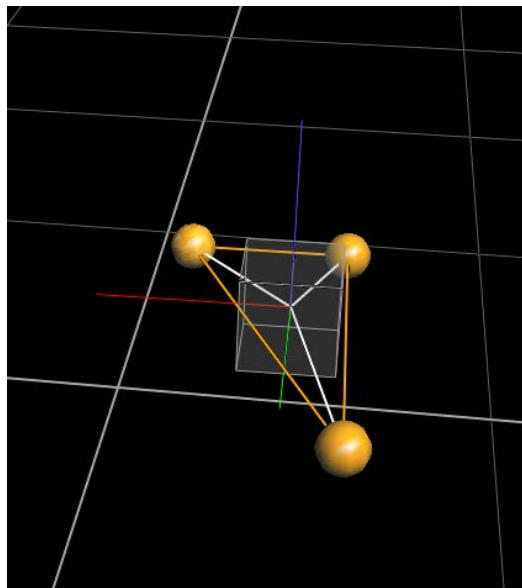


Figure 20. Tracker Program with the Traxxas Vehicle Model.

III. VEHICLE MODELING AND CALIBRATION

A. VEHICLE KINEMATIC MODEL

1. The Four-Wheeled Car

The model of a four-wheeled non-holonomic vehicle, with rear-wheel drive and front wheel (Ackerman) steering is found in Figure 21. The descriptions of the vehicle kinematics are found in (1) and (2). The commands for the actual vehicle are inputs of velocity, v , and steering angle, γ , but these inputs cannot be changed instantaneously. Therefore, different controls are used to solve the optimal control problem. These controls are the rate of velocity i.e. the acceleration, a , and the rate of steering, ω , which are found in (3).

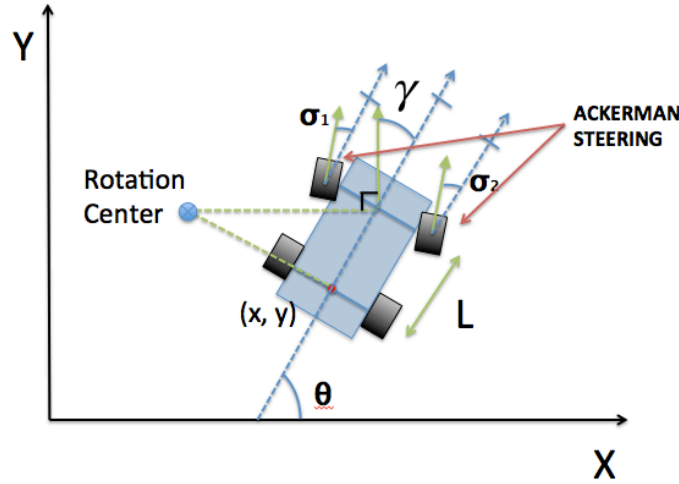


Figure 21. Model of a Front-Wheel Steering Vehicle.

$$\underline{X} = \begin{bmatrix} x \\ y \\ \theta \\ v \\ \gamma \end{bmatrix} \quad (1)$$

$$\dot{\underline{X}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \frac{v}{L} \tan(\gamma) \\ a \\ \omega \end{bmatrix} \quad (2)$$

$$\underline{u} = \begin{bmatrix} a \\ \omega \end{bmatrix} \quad (3)$$

2. The Need for Model Calibration

The states for the car are in units of meters per second for velocity and radians for the turning angle (γ). At the hardware level, however, the vehicle was controlled by a pulse width modulation to control the velocity and steering servos. Thus, calibration of the vehicle was necessary to develop a relationship to convert between the two sets of units to ensure the optimal trajectories are mapped to proper commands for the vehicle.

B. MODEL CALIBRATION

1. Turning Angle (γ) and Steering

The Ackerman steering allows the vehicle to rotate around a single point called the rotation center (see Figure 21). This rotation allows for a simple geometric relationship to determine the turning angle of a vehicle as shown in Figure 22. Driving the vehicle in a circle, the turning vector was tangent to the arc of the turning circle. The turning vector created an 90° angle with the turning radius and therefore the turning angle, γ , of the vehicle would be the same angle created by the triangle formed by the wheelbase (length L) and the distance from the center of rotation to the center of aft axle (length M):

$$\tan \gamma = L/M \quad (4)$$

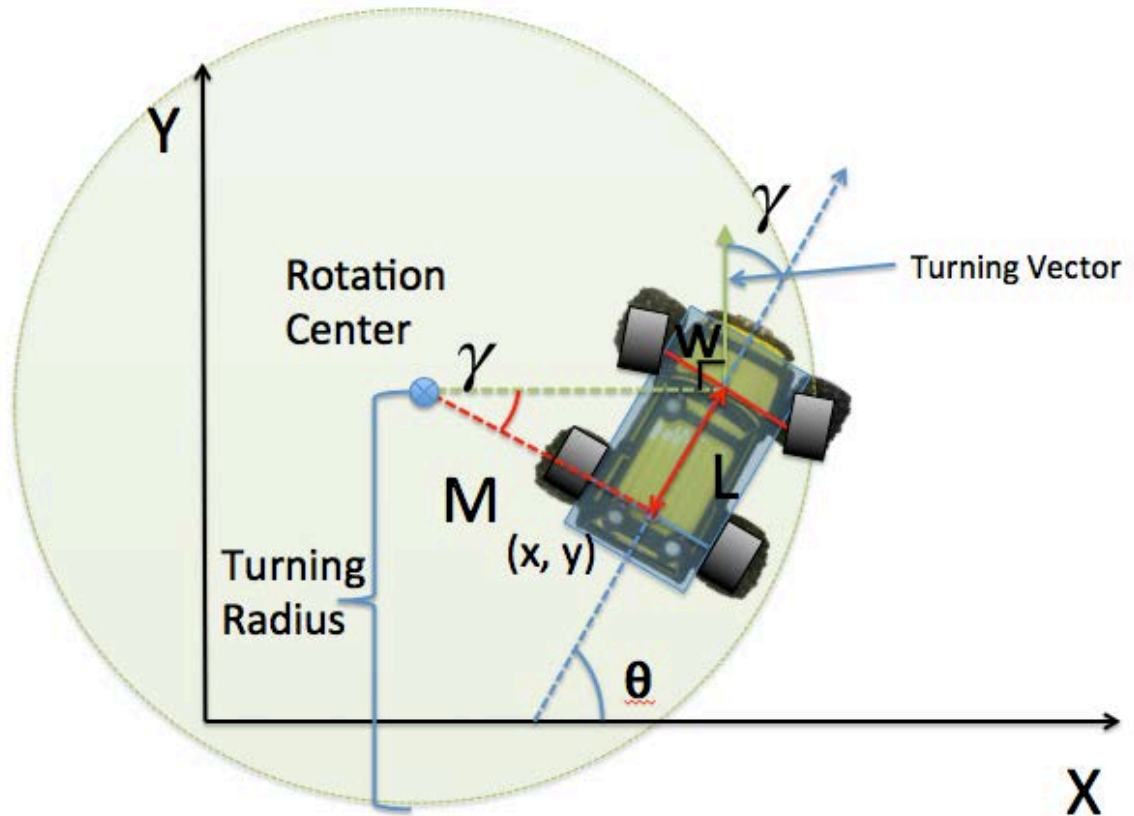


Figure 22. Evaluating the γ Angle.

Calibrating the vehicle's turning angle required the knowledge of the turning radius for the vehicle at various microsecond steering commands. The servos required inputs of microsecond pulse width ranging from 1000 to 2000 μs . Assumptions were made that a signal ranging from 1000 to 1499 μs commanded the steering left while the signal from 1501 to 2000 μs commanded the steering right. A final assumption was that a signal at 1500 μs would drive the vehicle straight. The command of 1000 μs turned the wheels to approximately $\sigma = 30^\circ$ off centerline. The angle σ (see Figure 21) has no direct relationship to the steering angle, γ , because the steering angle relates to the steering wheel of a car and not the orientation of the tires. The reason for this relationship between the steering and tires is the linkage of Ackerman Steering, which allows for the steering tires to each have their own angle. The γ angle, related to the steering wheel input, moves each tire by a different amount.

Calibration was required to determine the relationship of the vehicle's command input (rad) to the steering servo pulse width (μs). Calibrating the steering required the knowledge of the length of M in Figure 22. This was done by initially marking the ground at the center of the aft axle. A constant velocity with a constant steering command was applied to the vehicle until it completed a U-turn with respect to the starting position. The radius (length M) of the circle was calculated by measuring the diameter of the circle created by the U-turn (length D) as shown in Figure 23. Simple geometry allowed the γ angle to be determined using equation (4). For example, the command input of 2000 μs gave $M=1.03m$. The distance L, the length from the forward axle to the aft axle, is 0.337m. These measurements give a steering angle γ of 20° .

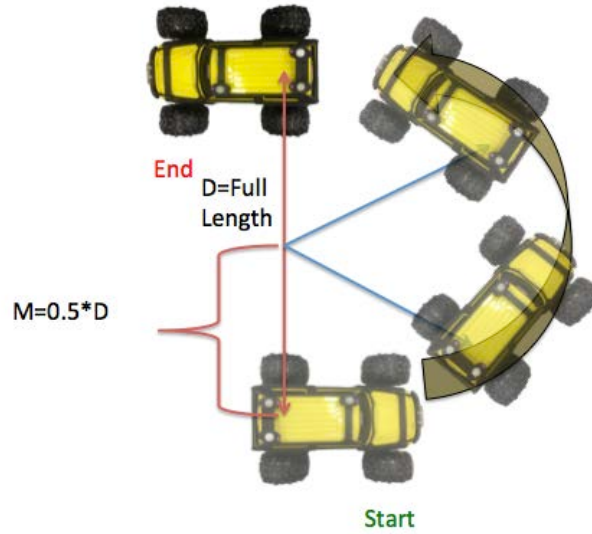


Figure 23. Visual Depiction of Determining Length of D.

Obtaining several values of the steering angle γ , a plot determined the linear relationship of γ to μs (see Figure 24). As stated earlier, to have the vehicle drive straight was assumed to require an input command of 1500 μs , however, the curve fit of the data in has the y intercept at 1431 μs . Thus, the difference between the nominal value of 1500 μs and the measured value of the intercept (1431 μs) represents the “steering trim” that needs to be added to the steering angle trajectory. The equation for mapping

the steering angle to steering pulse width is given by (5) where x is the desired steering angle and y is the pulse width command.

$$y = 1231x + 1431 \quad (5)$$

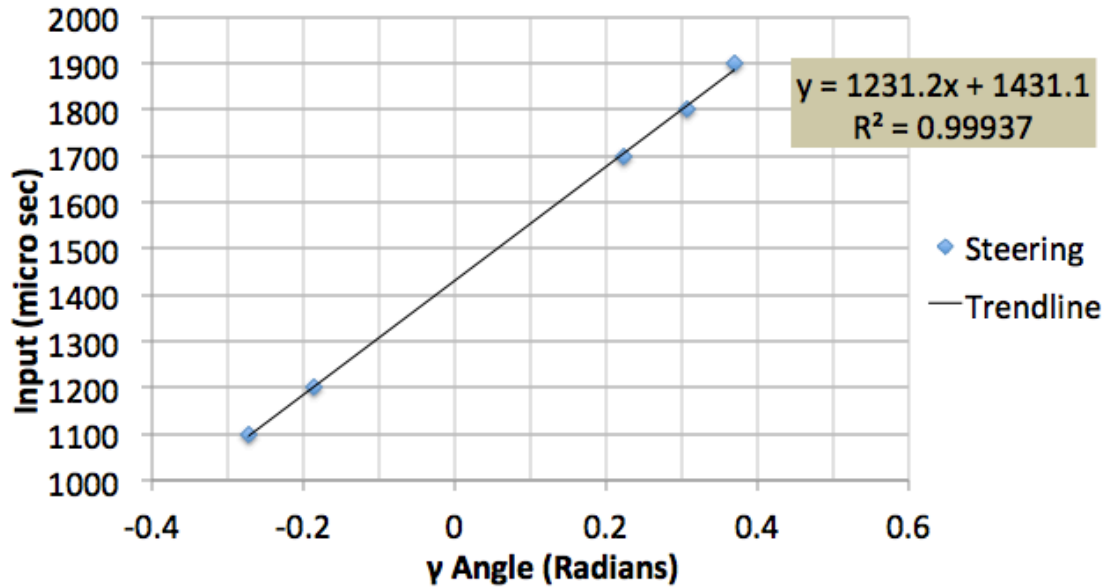


Figure 24. Experimental Calibration Curve for Steering Servos.

2. Vehicle Speed Calibration

To calibrate the vehicle's velocity in meters per second in terms of a pulse width for the HB-25 motor controller, the vehicle's speed for various constant input signals was captured using the VICON system. The Tracker program has an option to capture the velocity directly, but the readout was too noisy to determine the mean value of the vehicle's speed. Using a separate C-program to extract the position of the vehicle from VICON in terms of x , y , and z , at a rate of 100 frames per second, allowed for the magnitude of the velocity to be calculated. Due to the combination of a short distance to conduct the maneuvers involved for this thesis, and a maximum imposed acceleration of 0.045m/s^2 , the range of velocities required was between positive and negative one meter per second. This acceleration was chosen to allow for more realistic time to conduct the parking maneuvers. For example, a parking maneuver time greater than ten second was

desired, verses a maneuver conducted in less than three seconds since fast maneuvers caused the vehicle to slip on the laboratory floor. The various velocities were plotted on the graph in Figure 25.

The resulting velocity profile has a linear relationship in both the forward and reverse directions (see trendline equations (6) and (7) respectively). The slopes were slightly different, and the trendlines were not precisely lined up. There is a slight deadband (gap) from -0.1 to zero m/s in the reverse direction (see Figure 25). This gap could cause small errors in the final position of the vehicle's trajectory. The forward direction has no deadband so the overall impact on the velocity would be minimal. The equations for the forward and reverse trendlines are:

$$\text{Forward: } y = 111x + 1515 \quad (6)$$

$$\text{Reverse: } y = 105x + 1465 \quad (7)$$

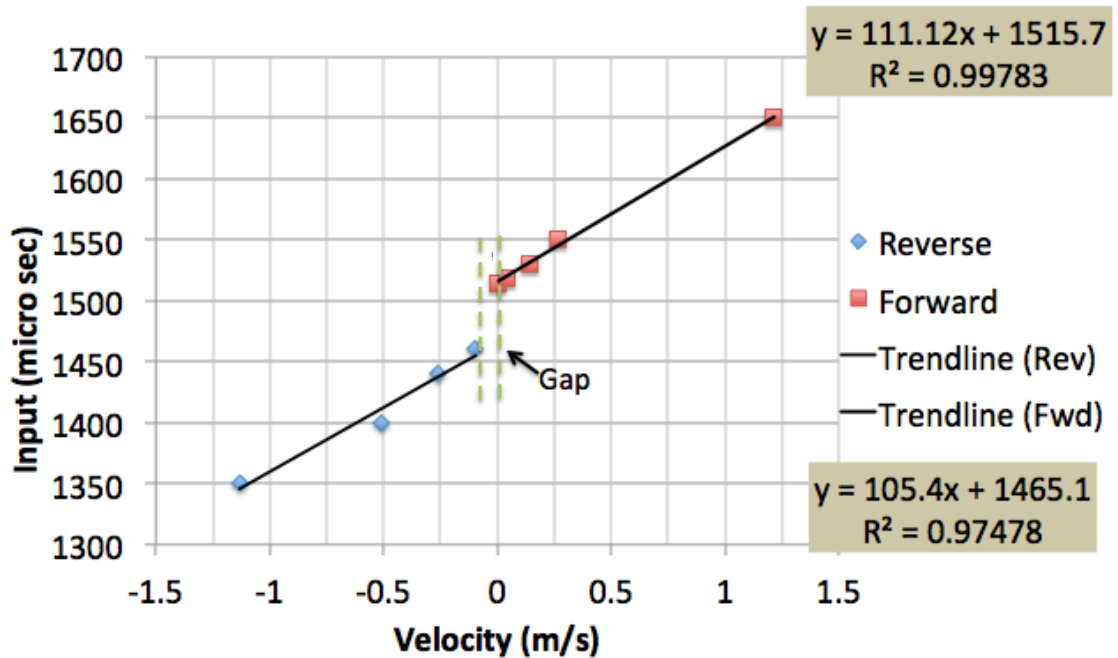


Figure 25. Experimental Velocity Profiles with Linear Trends.

3. Testing of Calibration

A trail test was conducted for a typical velocity trajectory to see how well the trajectory could be reproduced. In Figure 26 the test trajectory is compared with the trial velocity of the first attempt. The actual velocity did not match the required velocity profile for the maneuver. Slightly adjusting the value of the slope of the velocity profile allowed for better results during the maneuver. This is shown in Figure 27. The slope values were adjusted through trial and error, with resulting slope values of 170 for the forward direction and 145 for the reverse. Therefore, the “tuned” equations for the vehicle’s speed in meters/second to pulse width commands are equations (8) and (9). These were used in the experimental implementation of the time-optimal parking maneuvers.

$$\text{Forward: } y = 170x + 1515 \quad (8)$$

$$\text{Reverse: } y = 145x + 1465 \quad (9)$$

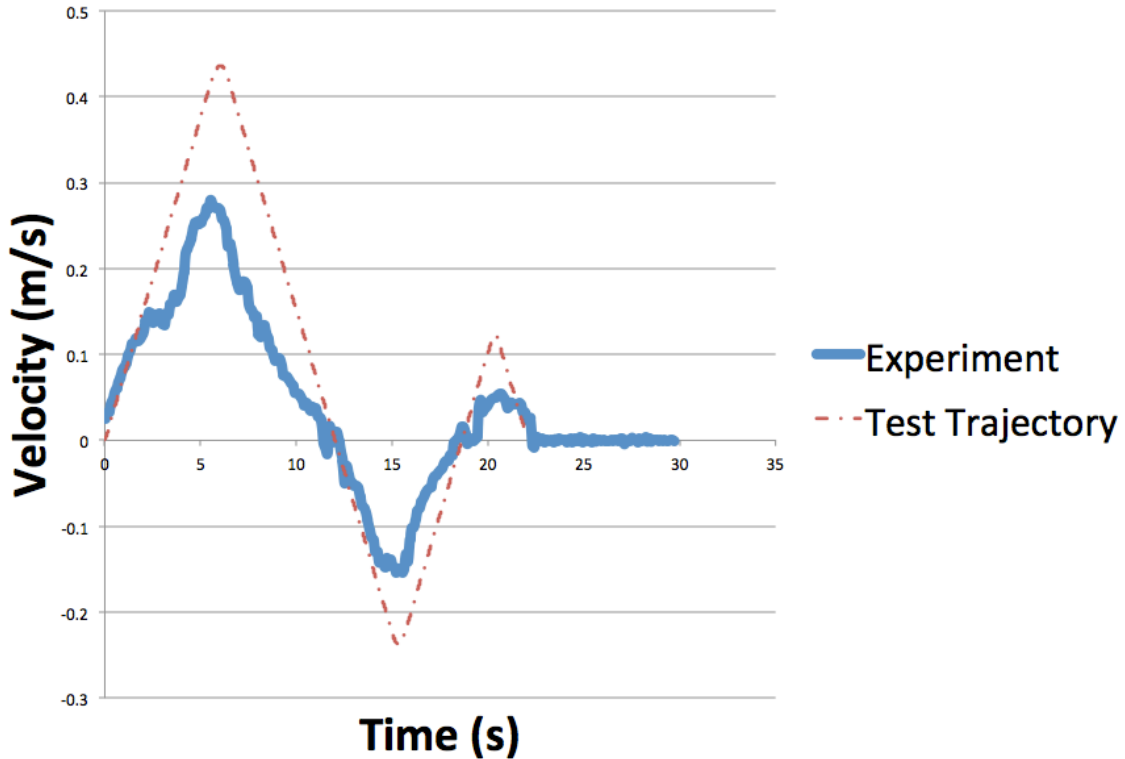


Figure 26. Velocity Profile Using Equation (6) and (7).

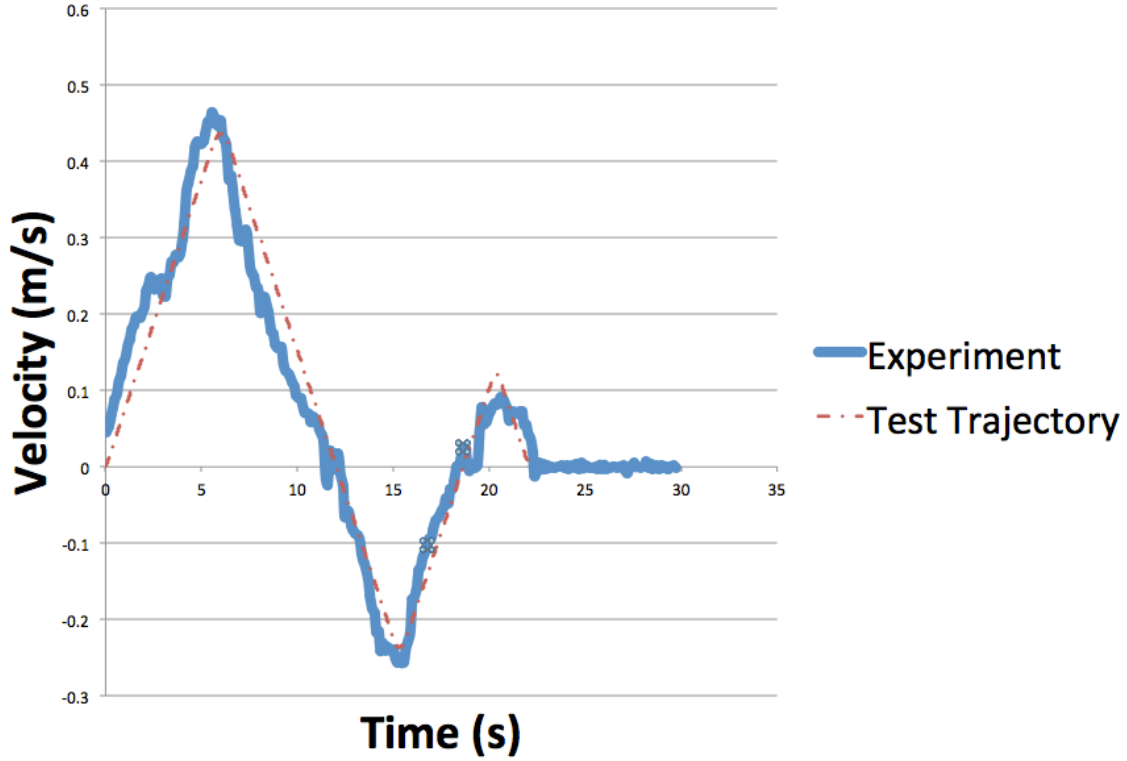


Figure 27. Velocity Profile After Tuning the Velocity Slope Using Equation (8) and (9).

During calibration, only the magnitude of the velocity was determined, but the experiment required a direction of the velocity as well. The magnitude of the velocity was still calculated for the experimental data, but the sign from the value of Δy was used to determine the sign of the velocity signal. The Δy value was used because the scenario had the majority of the movement in the y-direction.

The VICON system is a great tool to calibrate the vehicle, although the system does have some noise with an occasional data spike. Every ten data points were averaged to minimize the noise and eliminated the troublesome spikes.

The calibration of the steering was tested by computing the value of θ using equation (10), which was derived from the dynamic equation for \dot{x} . The angle θ was not calibrated directly, but was looked at as a possible future feedback output for the closed loop system. The experimental results are plotted against the test trajectory in Figure 28.

The two curves correlate well with one another indicating that the orientation of the vehicle can be properly controlled by the steering commands

$$\theta = a \cos\left(\frac{\dot{x}}{v}\right) \quad (10)$$

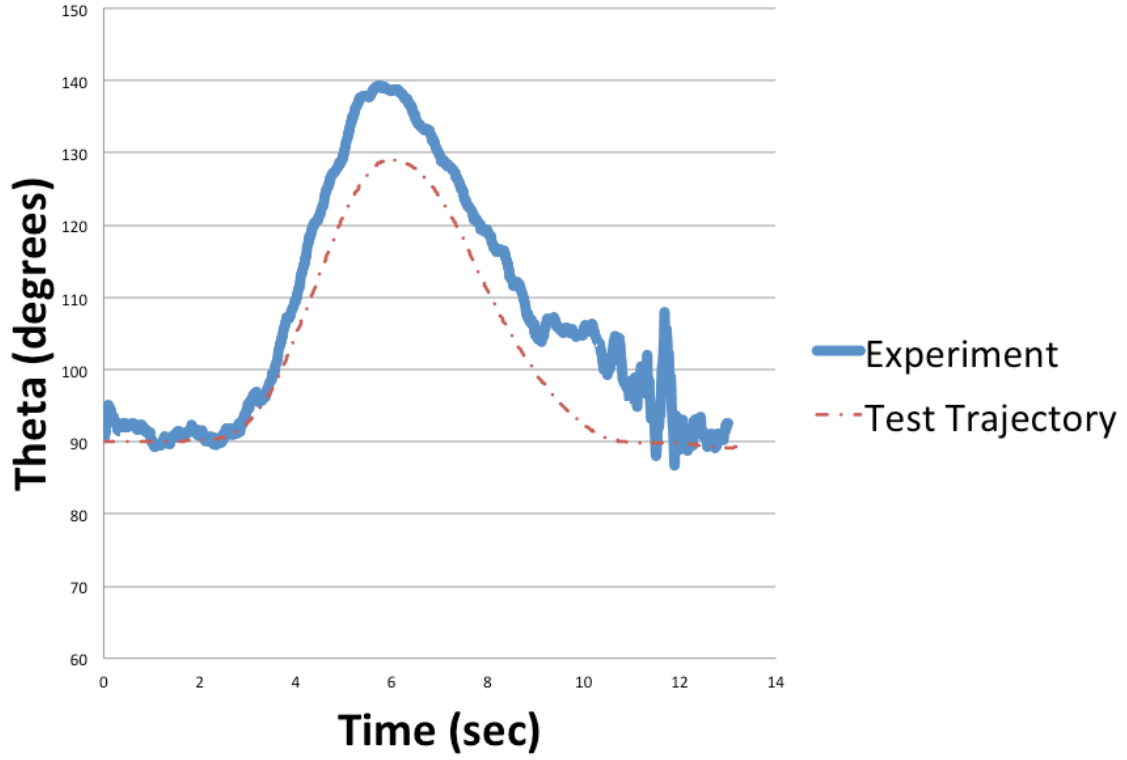


Figure 28. Steering Calibration Test for Controlling the Vehicle Orientation.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. A CANONICAL MOTION PLANNING PROBLEM

A. PARALLEL PARKING SCENARIO

The canonical scenario used for the demonstration of the autonomous motion-planning concept was an example of parallel parking. The challenges of parallel parking for non-holonomic vehicles seem obvious since automotive car companies started to offer automated solutions to assist in the process of parallel parking. The solutions offered by the automotive industry are, however not a fully autonomous system. They still have a human in the loop. The driver controls the gas and the brake pedal while computed inputs to the car assist in guiding the vehicle into the parallel parking spot.

Parallel parking is a useful scenario to apply to planetary rovers. Planetary rovers will be expected to conduct more routine operations and maneuvers in the future. In the near future, a mother ship with several planetary rovers with autonomous function of routine operations and maneuvering will be required. The parallel parking scenario is challenging for non-holonomic vehicles, but even with today's technology, it is difficult to control several land rovers conducting routine operations. The importance of this parallel parking scenario is that if a laboratory rover can conduct a challenging scenario similar to this, so too could a planetary rover on Mars.

1. Parallel Parking

The size of parallel parking spaces varies from city to city. Therefore, there were two approaches in which a parallel parking space for this thesis was determined. The first approach referred to the California driving handbook that stated to "look for a space about 3 feet longer than your vehicle to safely park in the space without striking another vehicle or object" [12]. A 2012 Honda Pilot was used as a test modeling reference. The Pilot is 15.5 feet long and adding the three feet as the California driving handbook recommends, the desired length of a parallel parking space for the Pilot would be 18.5 feet long. The Traxxas vehicle is 0.563 meters in length. The ratio of the Honda Pilot and the parking spot to the model of the vehicle yields a space 1.04 meters in length. The ratio of the vehicle to the length of the parking space gives a parking space that is 16%

longer. The problem with this solution was the three feet as it applied to all sizes of vehicle and did not take into account the turning capability of the vehicle.

Ultimately it does not matter how big the space allotted by different cities is, only how much space is left between vehicles when parking. Three case studies each with three scenarios will be evaluated in this thesis. Case 1 will be the traditional approach of backing in, case two will have the vehicle and the empty space aligned next to each other, and case three will allow the vehicle to attempt to drive forward directly into the space. With each case there were three scenarios tested, first with no cars or obstacles, second with cars parked in front and back of the parking space allowing ideal “perfect” spacing, and finally a solution to park in a minimal space.

The second approach of “perfect” parallel parking used Simon Blackburn’s paper on the “The Geometry of Perfect Parking” [13]. This paper determined the desired length of a parallel parking space for backing into the parking space without having to move in a forward direction. Using equation (11), from Blackburn’s paper, the space required for a 2012 Honda Pilot was 18 feet. This is about 6 inches shorter than the value recommended by the California Driver’s Handbook.

The “perfect” spot to parallel park is shown in Figure 29. To solve for the additional distance required to park for a vehicle, $|AH|$, the following lengths are required $|EX|=r$, $|EF|=l$, $|AE|=k$, and $|GH|=w$. The variable r represents the turning radius for the outside tire, l represents the wheelbase length, k represents the distance from the front tire to the front of the car, and w is the width of the parked car. These dimension are shown in Figure 30.

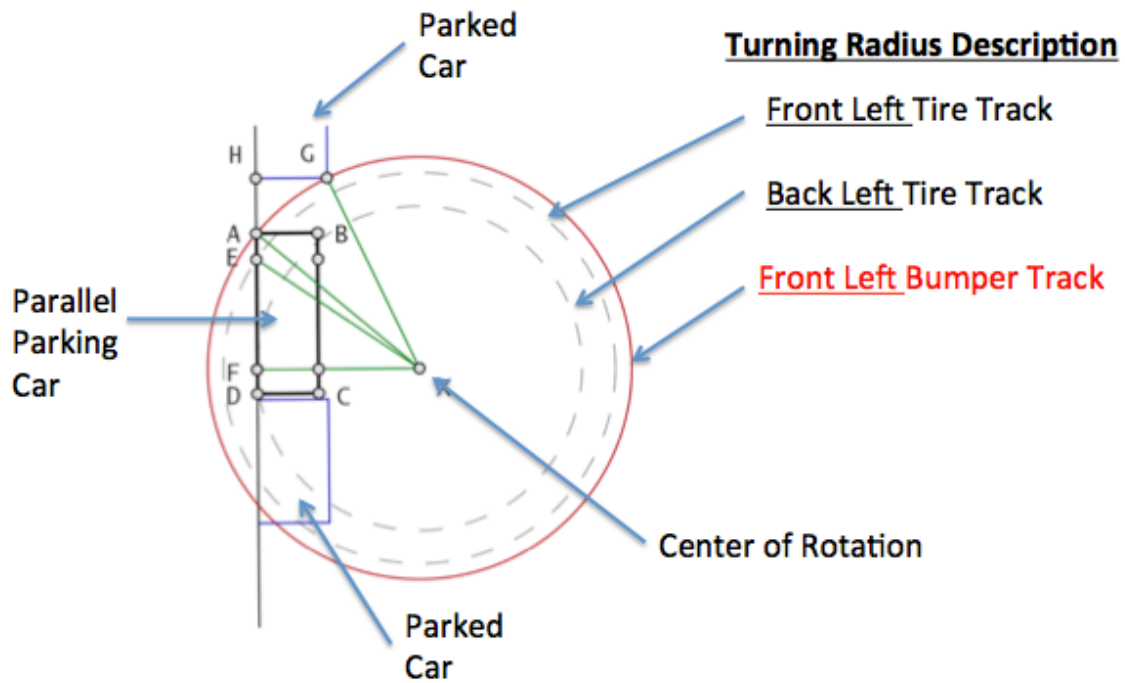


Figure 29. Schematic of the Geometry of Perfect Parallel Parking after [13].

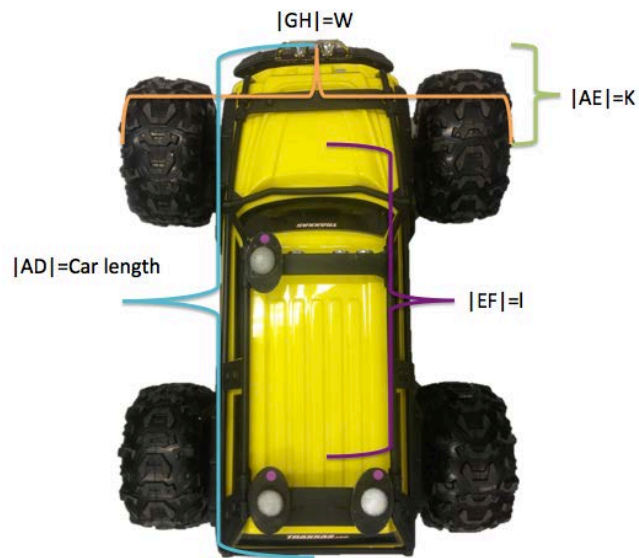


Figure 30. Schematic of Lengths for a Geometrically Perfect Parallel Parking.

The length of the ideal parallel parking as per Figure 29 is given by the following equation:

$$|DH|=|DA| + \sqrt{(r^2 - l^2) + (l + k)^2 - \left(\sqrt{r^2 - l^2} - w\right)^2} - l - k \quad (11)$$

The turning radius, r , for the vehicle was not a value that could be measured directly; therefore, the turning radius was estimated utilizing equation (12). Due to the vehicle's width, w , of 0.5m and a steering angle, γ , of 20° , the estimated value of the turning radius is $r = 1.35\text{m}$.

$$r = \frac{w}{2} + \frac{l}{\sin \gamma} \quad (12)$$

Using the turning radius of 1.35m, along with $k=0.25\text{m}$, $w=0.5$, $l=0.377$, and $|AD|=0.877$ the “perfect” parallel parking spot using equation (11) becomes 1.45m for the Traxxas vehicle. The perfect parallel parking spot for this scenario was 40% larger than the vehicle.

2. Parallel Parking in Minimum-time

Equations (13) through (16), describe the problem formulation for an optimal control solution for minimum time parallel parking. Equation (14) was the dynamics of the model of the vehicle. Equation (15) describes the controls for the non-holonomic vehicle. The two controls were acceleration, a , and the steering rate, ω . The steering rate represents the rate at which γ changes with respect to time. Equations (16) identify the initial conditions and final conditions, of vehicle in case 3, which is the case where the vehicle attempts to drive into the parking space. Equations (17) describe the upper and lower bounds of the controls of the vehicle.

$$J[\underline{X}(\cdot), u(\cdot), t_f] = t_f \quad (13)$$

$$\dot{\underline{X}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \frac{v}{L} \tan(\gamma) \\ a \\ \omega \end{bmatrix} \quad (14)$$

$$\underline{u} = [a, \omega] \quad (15)$$

$$\begin{aligned}
(x_o, y_o, \theta_o, v_o, \gamma_o) &= (4.6, 4.5, \pi/2, 0, 0) \\
(x_f, y_f, \theta_f, v_f, \gamma_f) &= (5.5, 6.4, \pi/2, 0, 0) \\
t_o &= 0
\end{aligned} \tag{16}$$

$$\begin{aligned}
-0.045 &\leq a \leq 0.045 \\
-0.224 &\leq \omega \leq 0.224
\end{aligned} \tag{17}$$

3. Validating the Minimum-Time Problem

To validate the results of the solution to the minimum-time problem, it was necessary to define the Hamiltonian, which is given in equation (18):

$$H(\underline{x}, \underline{u}) = \lambda_x v \cos(\theta) + \lambda_y v \sin(\theta) + \lambda_\theta \frac{v}{L} \tan(\gamma) + \lambda_v a + \lambda_\gamma \omega \tag{18}$$

Three equations are analyzed at for validity of an optimal solution. First is the Euler-Lagrange equation.

$$\begin{aligned}
\frac{\partial H}{\partial a} &= \lambda_v = 0 \\
\frac{\partial H}{\partial \omega} &= \lambda_\gamma = 0
\end{aligned} \tag{19}$$

The control variables are not explicitly present in (19), so the Hamiltonian Minimization Control (HMC) is applied.

$$(HMC) \begin{cases} \text{Minimize} & H(\lambda, x, u) \\ \text{Subject to} & u^L \leq u \leq u^U \end{cases} \tag{20}$$

Application of the HMC gives the switching functions that allow the control values to be determined. For example, the switching function for the acceleration is,

$$\begin{aligned}
S_a = \lambda_v &\rightarrow \lambda_v > 0 & a &= a_{\min} \\
&\lambda_v < 0 & a &= a_{\max} \\
&\lambda_v = 0 & a_{\min} &\leq a \leq a_{\max}
\end{aligned} \tag{21}$$

Next, the adjoint equations are evaluated. The adjoint equations gave useful information for the co-states of x and y . Since the time-derivatives of λ_x and λ_y are zero, the values of the co-states will be constant. Their particular values will vary, case to case.

$$\begin{aligned}
\dot{\lambda}_x(t) &= \frac{-\partial H}{\partial x} = 0 \rightarrow \lambda_x = A \\
\dot{\lambda}_y(t) &= \frac{-\partial H}{\partial y} = 0 \rightarrow \lambda_y = B \\
\dot{\lambda}_\theta(t) &= \frac{-\partial H}{\partial \theta} = \lambda_x v \sin \theta - \lambda_y v \cos \theta \\
\dot{\lambda}_v(t) &= \frac{-\partial H}{\partial v} = -\lambda_x \cos \theta - \lambda_y \sin \theta - \frac{1}{2} \lambda_\theta \tan \gamma \\
\dot{\lambda}_\gamma(t) &= \frac{-\partial H}{\partial \gamma} = -\lambda_x \frac{V}{L} \sec^2 \gamma
\end{aligned} \tag{22}$$

The third condition that was evaluated was the transversality condition. The specific form for \bar{E} is shown in (23). The application of the transversality condition equation (24) does not give any useful information.

$$\bar{E}(v, \underline{x}(t_f)) = v_x(x_f - x_o) + v_y(y_f - y_o) + v_\theta(\theta_f - \theta_o) + v_v(v_f - v_o) + v_\gamma(\gamma_f - \gamma_o) \tag{23}$$

$$\begin{aligned}
\lambda_x(t_f) &= v_x \\
\lambda_y(t_f) &= v_y \\
\lambda_\theta(t_f) &= v_\theta \\
\lambda_v(t_f) &= v_v \\
\lambda_\gamma(t_f) &= v_\gamma
\end{aligned} \tag{24}$$

To validate the solution of the time-minimum problem, the results of the solution must comply with the following:

- (i) $\lambda_x = \text{constant}$
- (ii) $\lambda_y = \text{constant}$
- (iii) The value of the Hamiltonian for a minimum time problem is -1.

4. Using the DIDO Tool

The DIDO program consists of the formulation of the problem to include the initial and final state, kinematics and dynamics of the problem. In addition there was an option to add a path function to accommodate obstacles. As mentioned, the path function permits restriction of distances to hazards and obstacles (details in next section). In addition, the DIDO tool represents the vehicle as a single geometric point at each time step, which leads to the solution having “no volume”.

5. Modeling Obstacles

Obstacles for the scenario consist of two objects (parked cars) and a barrier (curb). In the path function for DIDO, the planetary rover and the objects were modeled as a combination of two circles, while the barrier was modeled as single line (see Figure 31 and Figure 32). The circle was chosen for its simplicity in modeling the shape of the obstacle and minimizing the need for complex equations during the computation of the optimal solution.

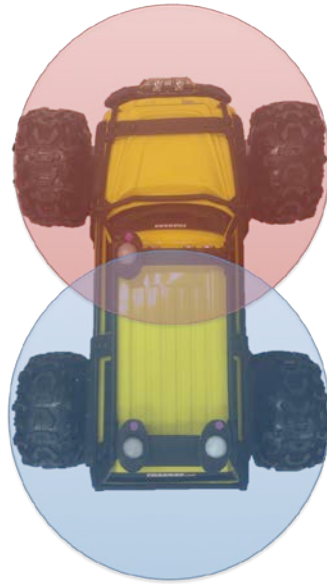


Figure 31. Vehicle Represented in DIDO With Two Circles.

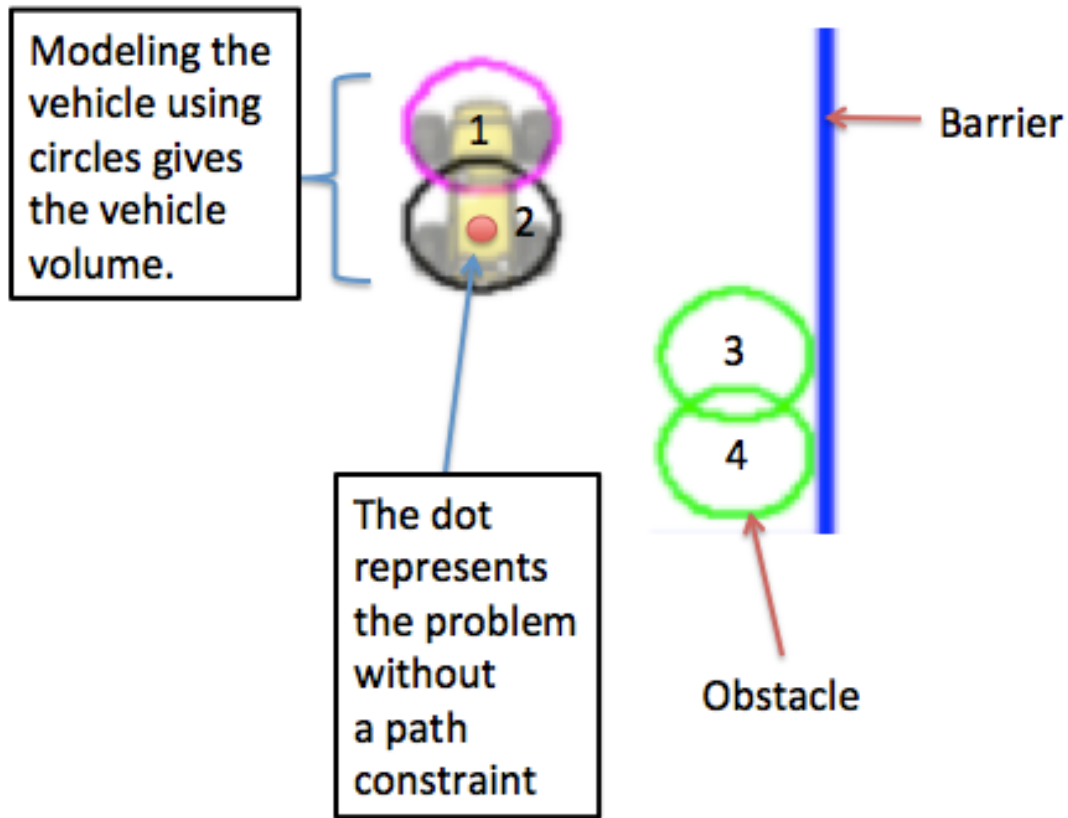


Figure 32. Model of Vehicle, Obstacle, and Barrier

To successfully avoid obstacle hazards, the path function required that a minimum length be kept between two modeled objects. Example of an equation for minimum distance (D_m) between two circles is found in equation (25) with a visual representation shown in Figure 33.

$$\begin{aligned}
 r_1^2 &= x_1^2 + y_1^2 \\
 r_3^2 &= x_3^2 + y_3^2 \\
 D_m &= r_1^2 + r_3^2
 \end{aligned}
 \tag{25}$$

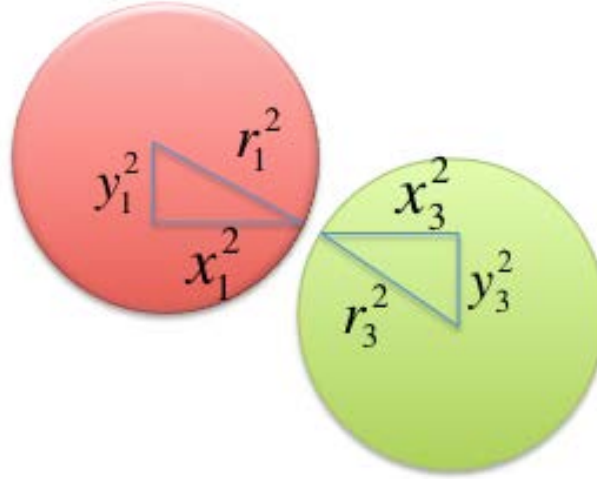


Figure 33. Minimum Distance Between Two Circles.

B. TRAJECTORY VALIDATION

The solution from any tool needs to be validated. To validate the optimal control solution from DIDO, the following three tests were carried out: First was the Hamiltonian test, followed by the Costates test and finally the feasibility test.

1. Hamiltonian

The Hamiltonian must have a constant value of negative one for the solution to be minimum time (see Figure 34).

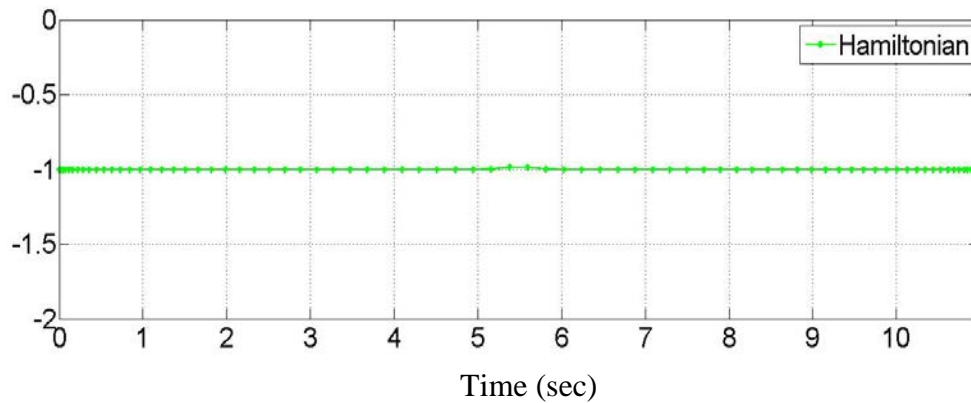


Figure 34. Example of the Hamiltonian With a Constant Value of Negative One During the Entire Commanded Trajectory (Problem given in 13 to 16).

2. Costates

The results of the analysis of the adjoint equations lead to the values of λ_x and λ_y must be constant. As seen in Figure 35, the values of the costates (λ_x and λ_y) are indeed constant for the problem (13) to (16).

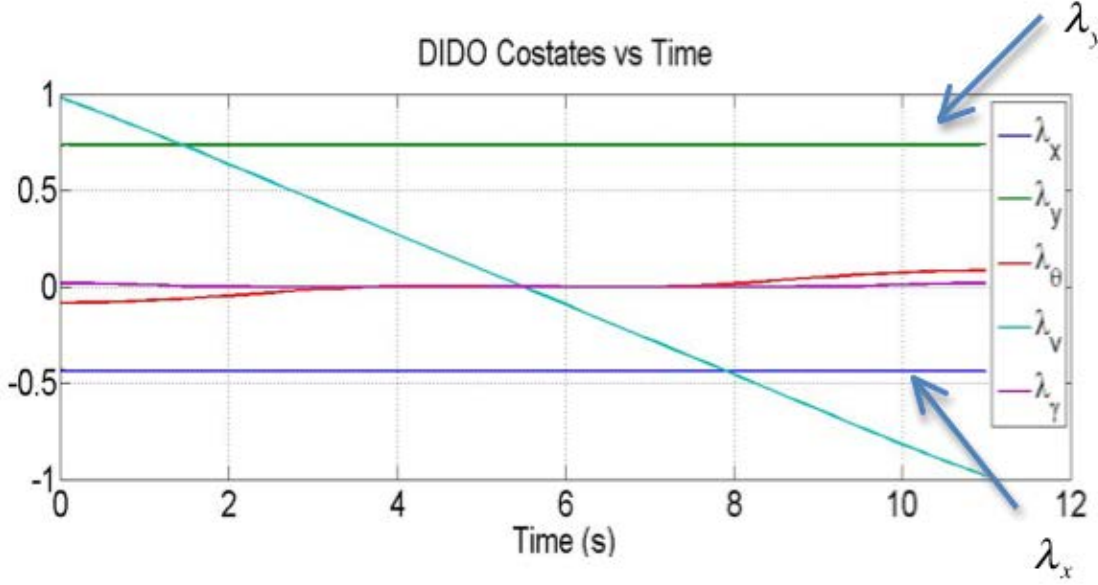


Figure 35. Graph of Costates.

3. Feasibility Test

A feasibility test checks if the solution of the commands moves the vehicle in the desired way. Propagation of the dynamics is a method used to test the feasibility of the solution. The propagation tool used was ODE45. The results of the feasibility test for the states are found in Figure 36.

4. The Path Function Affects the Hamiltonian and the Costates

The Hamiltonian and the Costates, λ_x and λ_y are constants if the solution is optimal. However, the addition of the path function can effect the values of the Hamiltonian and costates if the objects become close to one another. The effects on the

costates values can be seen when the optimal solution makes it necessary to have the vehicle pass closely to an obstacle. Figure 37 shows these effects when the vehicle become too close to an object and then the curb. To verify the optimization of the solution in the case, the Hamiltonian first needs to be augmented to include the path constant. However, in the absence of this, the value of Hamiltonian close to -1 can still be used in a quick check on the validity of the solution.

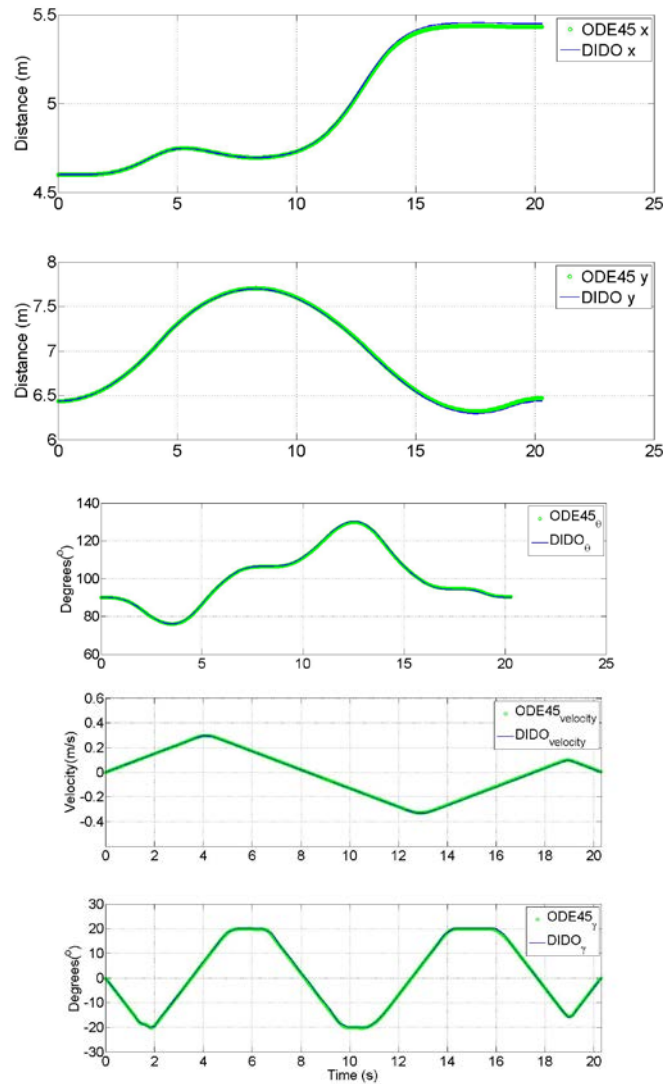


Figure 36. Comparison of the Optimal Solution to the Propagated Solution from ODE 45 (Problem in 13 to 16).

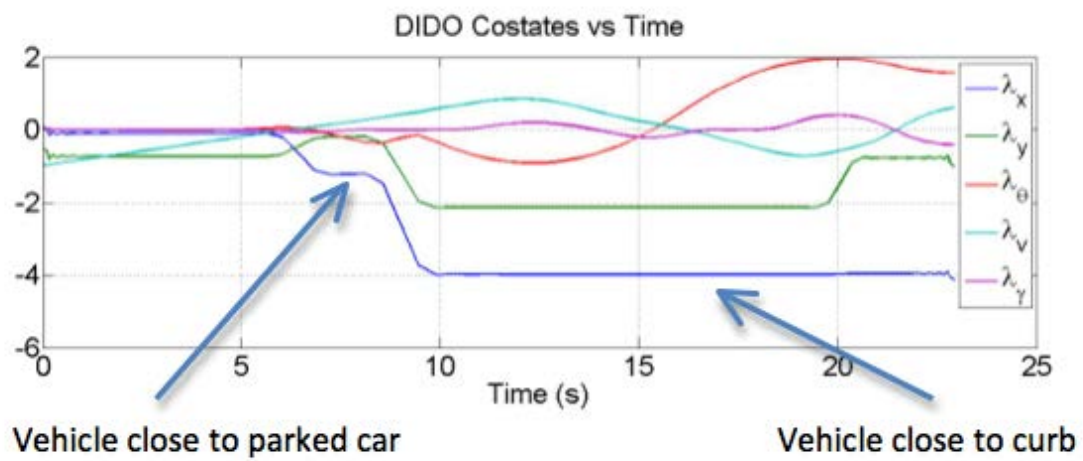


Figure 37. The Effect of the Path Function on Costates.

V. EXPERIMENTAL RESULTS

A. INTRODUCTIONS

There are many scenarios that can be tested for parking in a parallel spot. Utilizing DIDO to obtain the optimal solution for minimum time to park, there will be three cases examined. The first case was the traditional approach of backing into the space. The second case was if the car started right next to the space. In the third case, the car has the option to pull into the space while driving forward. The cases are all shown schematically in Figure 38.

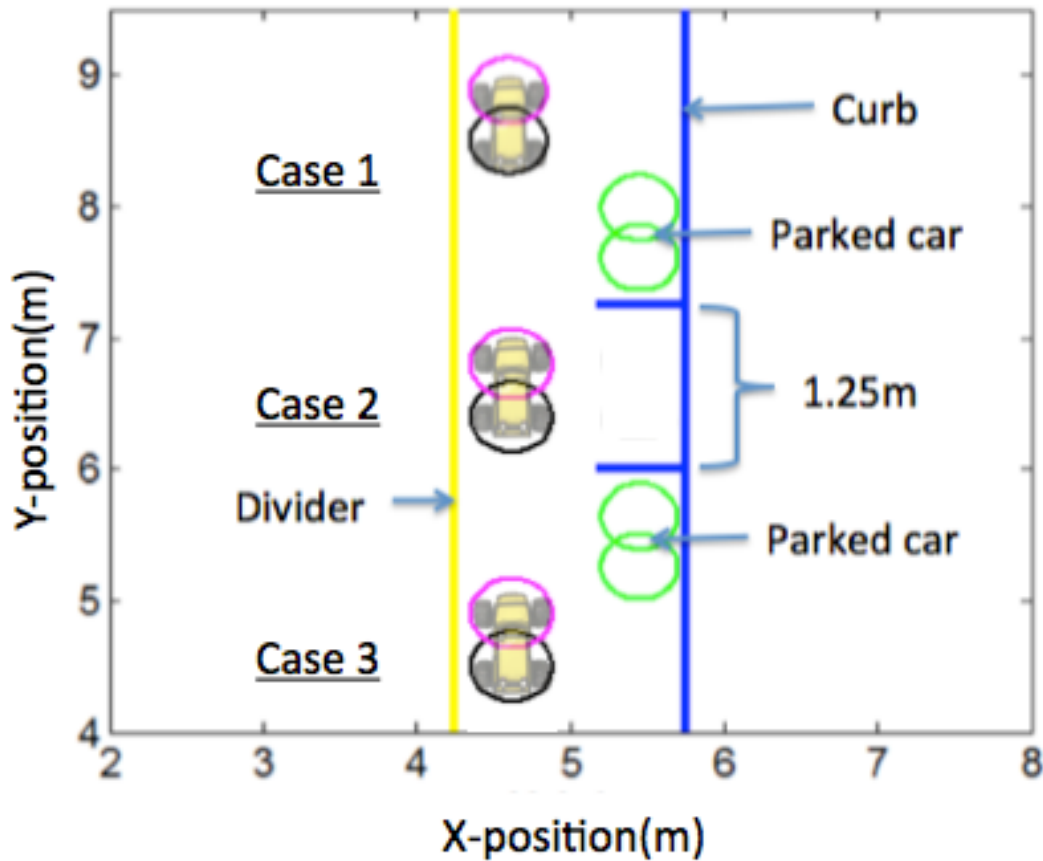


Figure 38. Schematic of the Initial Positions of the Three Parallel Parking Cases.

Within each of the three cases there will be three scenarios tested. The first scenario will be with no cars in the adjacent parking spaces. The second will have vehicles in both the forward and the aft parking spaces with the optimal amount of space as referred to in the “perfect” parking space scenario (i.e. 1.47m). The third scenario will be with the same vehicles, but with only the minimal amount of room on either side of the parking space (i.e. 1.25m).

B. TIME-OPTIMAL SOLUTIONS

Figures 39 through 41 show the results of all three initial conditions starting with the first scenario with no cars, second scenario with cars parked with an ideal amount of space, and the third scenario with cars parked with minimal space for a time optimal problem. The complete state and costate trajectories for each solution can be found in the Appendix.

The first scenario for all three cases had no parked cars near the parking spot. The results do not reveal anything surprising for case 1. The car simply backs into the spot. It is interesting that the vehicle in case 2, the minimum-time solution, had the vehicle move laterally half way to the spot by moving forward. The vehicle then reverses in to the spot, similar to case 1. Case 3 accomplished the task without a parallel parking move. The car simply drove forward in to the spot. This is similar standard to what human would do if trying to park in the minimum amount of time.

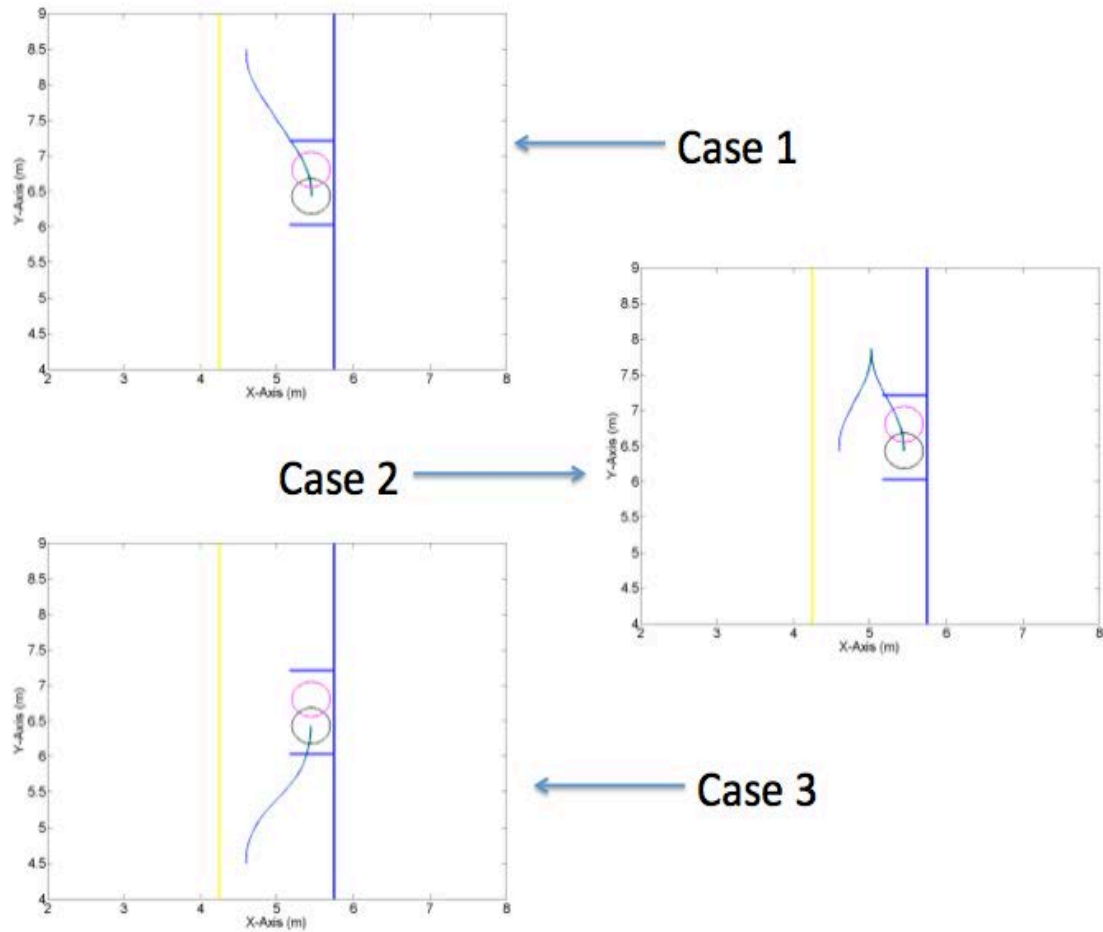


Figure 39. Optimal Trajectories for Scenario With No Cars

In Figure 40 the second scenario with the “perfect” amount of parking space with two parked cars is shown. No surprising maneuver was observed in the results of case 1. Case 2, followed the first scenario as to positioning the vehicle in a manner that mimics case 1 as it backs into the space. However, the forward motion is different that when no cars are present. Case 3 could not drive straight into the parking space like the earlier scenario. The vehicle first passes the parking space, but still tries to drive into the space as much as possible in order to position itself in a position very similar to case 1. Then the vehicle begins to reverse in to the parking spot in a motion that looks very much like the case 1.

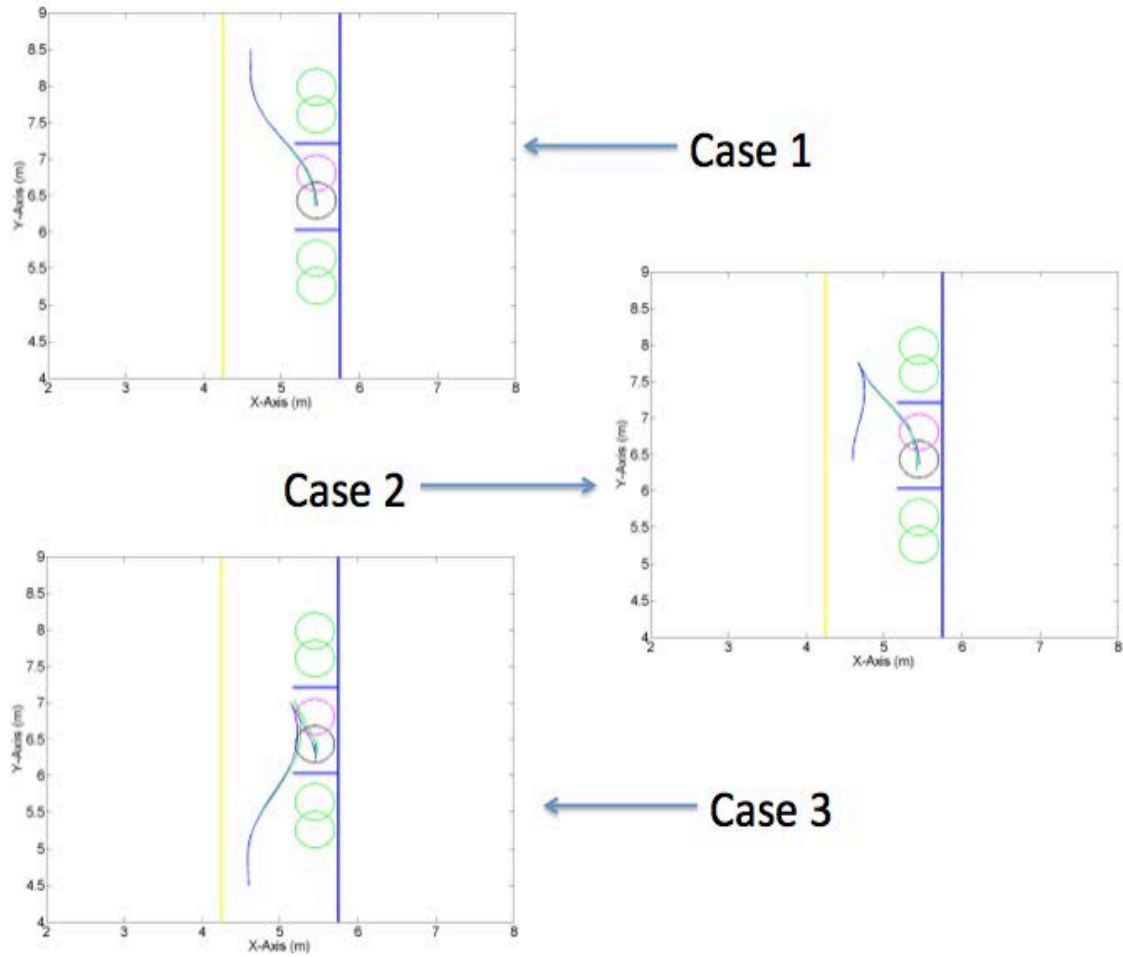


Figure 40. Optimal Trajectories for Scenario With Ideal Spacing

The scenario with minimal parking space is shown in Figure 41. The traditional parallel parking maneuver (backing in) hardly changes. Case 2 had very similar results as case 2 in scenario two. Case 3 conducted a similar maneuver as the previous scenario, but this time the vehicle had to drive slightly past the parking spot because there was less distance available in the space.

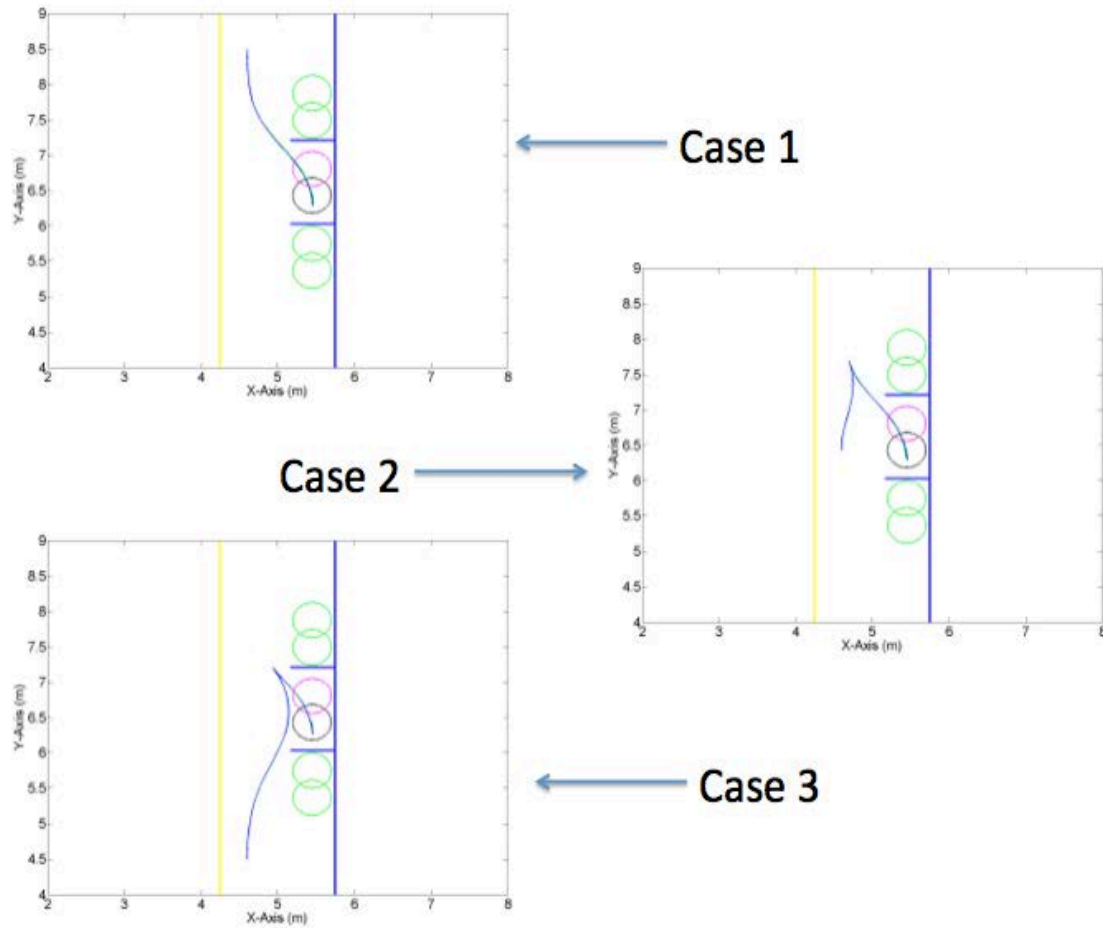


Figure 41. Optimal Trajectories for Scenario With Minimal Spacing

The optimal times to parallel park for each scenario are listed in Table 1.

	Optimal Trajectory Time
<u>Scenario One</u>	
Case 1	11.0 sec
Case 2	18.0 sec
Case 3	11.0 sec
<u>Scenario Two</u>	
Case 1	11.1 sec
Case 2	19.5 sec
Case 3	22.0 sec
<u>Scenario Three</u>	
Case 1	14.1 sec
Case 2	21.0 sec
Case 3	23.0 sec

Table 1. Optimal Times to Parallel Park.

C. IMPLEMENTATION

The optimal maneuvers were all implemented in experiments using the setup described in Chapter II. Figures 42 and 43 summarize the experimental results of two of the scenarios. Figure 42 shows the results of the minimal parking space for a time optimal solution. Figure 43 shows the results for the “perfect” space. The results of the data show that although the vehicle can successfully park, the control logic ideally needs to be a closed-loop solution. In Figure 42 the minimal space experimental data had one case where the vehicle very close to the intended trajectory. The reason for this result was the case in which the vehicle was tuned and calibrated. All other experiments were conducted shortly after the batteries were replaced. Therefore, the vehicle had more power available which allowed the vehicle to travel a greater distance in the open-loop. Closing the control loop, which is beyond the scope of this thesis, would be required to produce more accurate results.

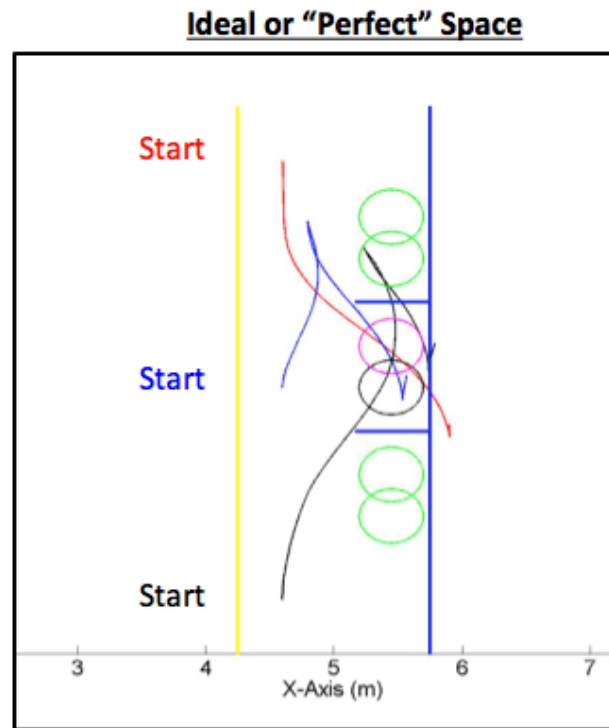


Figure 42. Experimental Results of All Three Cases for the Ideal Parking Space.

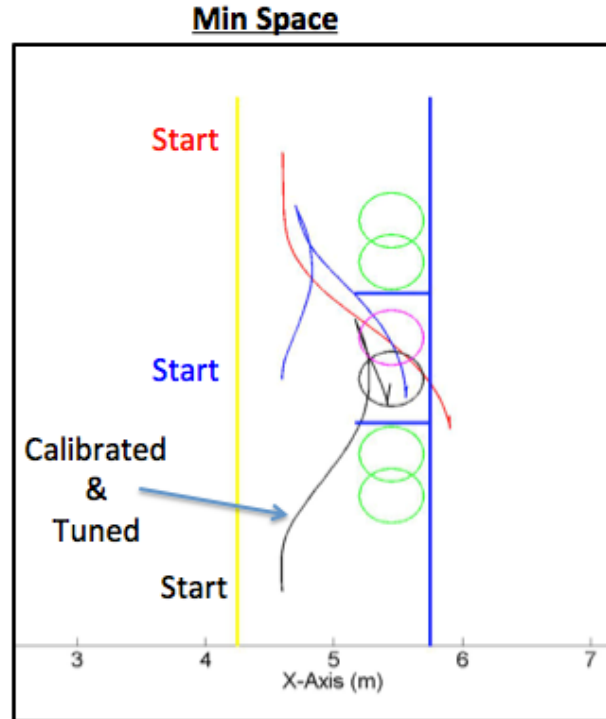


Figure 43. Experimental Results of All Three Cases for the Minimal Parking Space.

The experimental times were computed with a stopwatch and visual cues of the vehicle maneuvering. Table 2 shows the experimental results are in agreement with the optimal solution.

Scenario Three	Optimal Trajectory Time	Experimental Trajectory Time
Case 1	14.1 sec	13.4 sec
Case 2	21.0 sec	20.6 sec
Case 3	23.0 sec	24.4 sec

Table 2. Experimental Time for Parallel Parking.

Capturing video footage and VICON system could not be done at the same time, so the images discussed below were captured prior to the collection of data with the VICON system. Figures 44 through 46 show a series of still pictures that were taken from video of the experimental trajectories for scenario three. The video images were taken after calibration of the vehicle and prior to the changing the batteries. The blue tape in the parking spot was the target of the center aft axle. The still images show how the vehicle

moves in order to parallel park in minimum-time. The final parking position is somewhat inaccurate, however, due to the open-loop implantation. Closing the loop would improve this. Nonetheless, in each case the vehicle can successfully park.

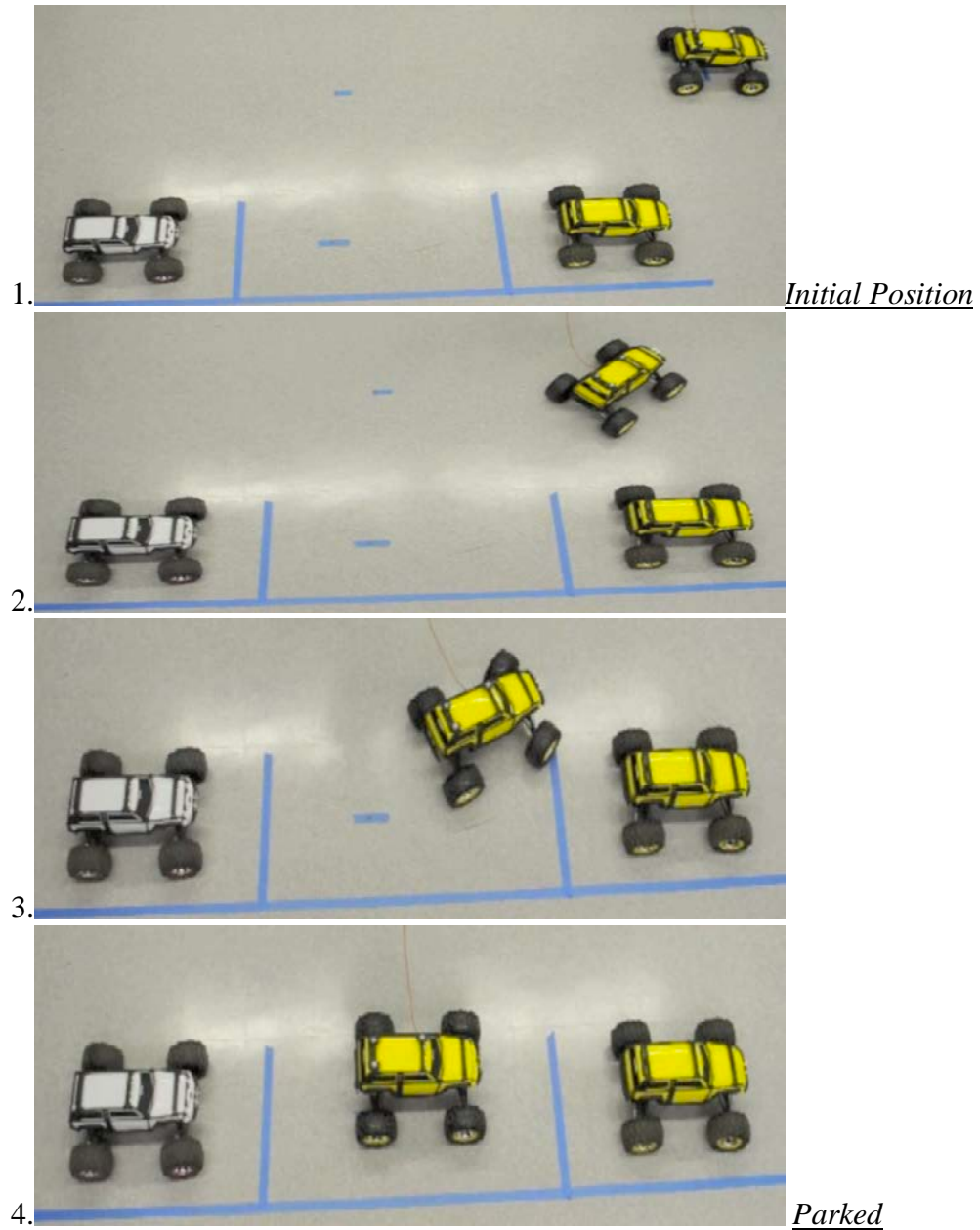


Figure 44. Still Photos of the Experiment for Scenario Three: Case 1.

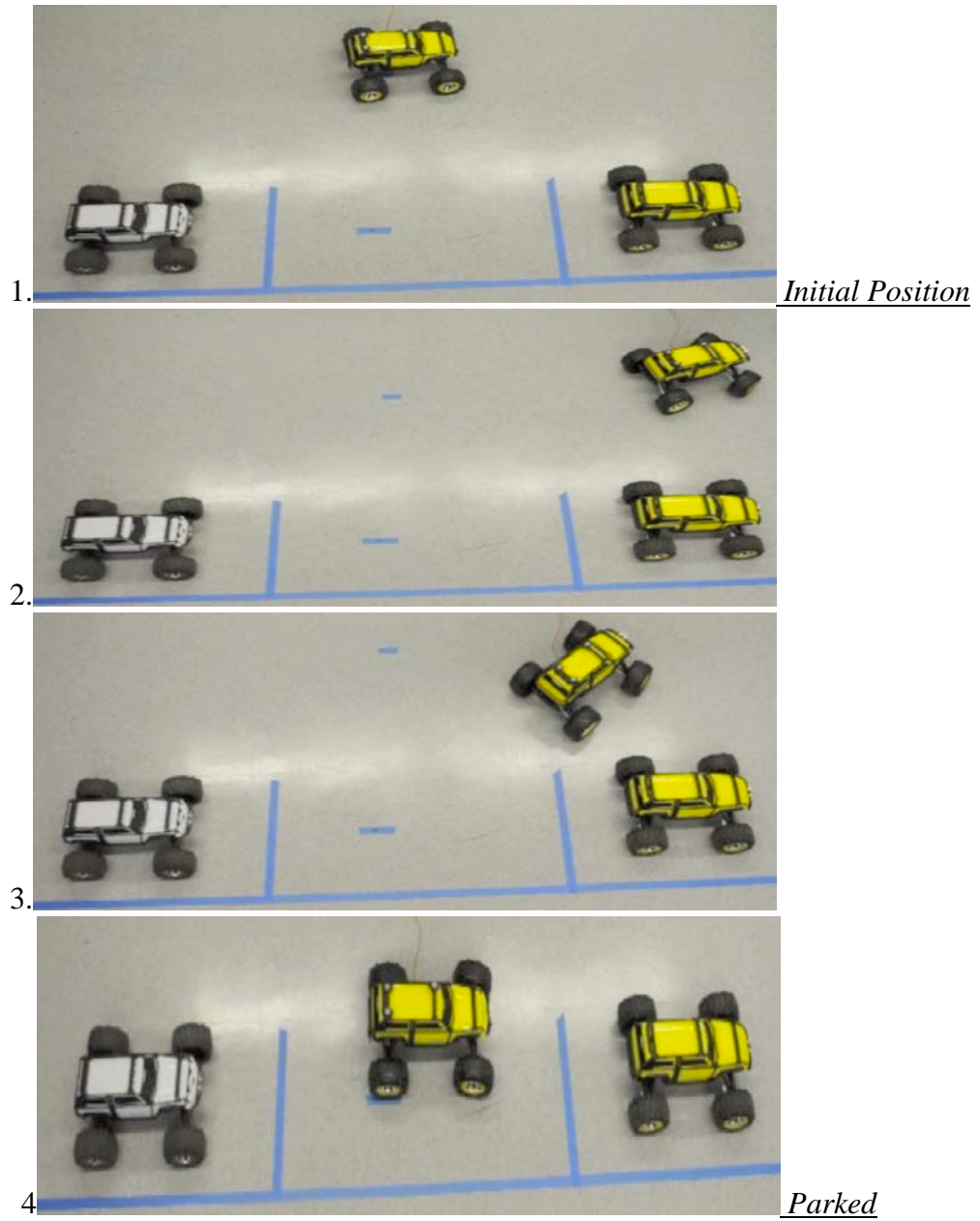


Figure 45. Still Photos of the Experiment for Scenario Three: Case 2.

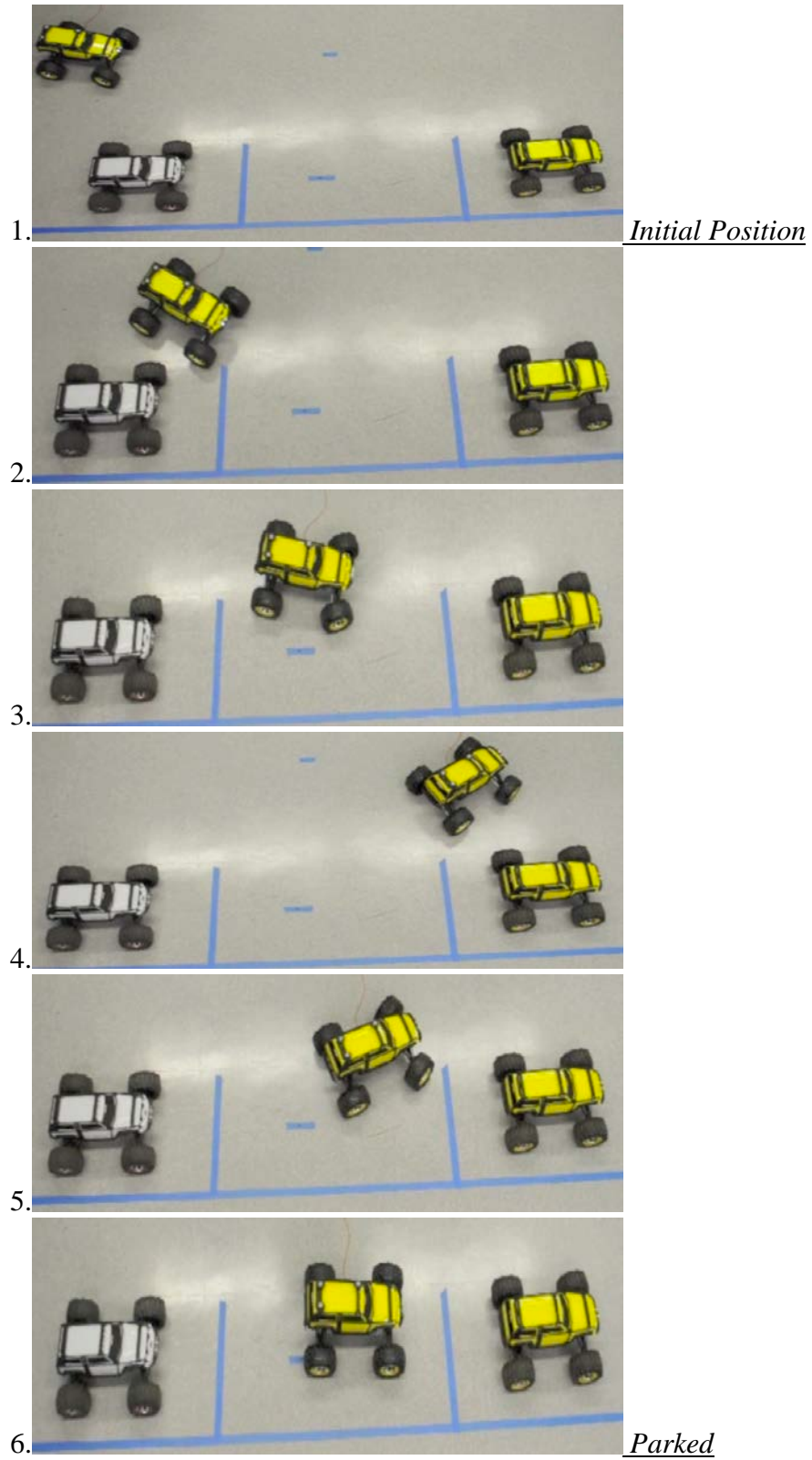


Figure 46. Still Photos of the Experiment for Scenario Three: Case 3.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSION

A Traxxas remote controlled vehicle was adapted and modified to accept an optimal trajectory plan given as control inputs of velocity and steering angles. The controller received the trajectory inputs from a computer running DIDO. The result from DIDO was a solution to a minimum-time problem that allowed a generic non-holonomic vehicle to successfully parallel park from different initial locations. Utilizing the VICON motion camera system, the experimental vehicle's trajectory was captured and compared to the optimal trajectory.

The vehicle performed very well, error of up to 3% in the y-direction and less than 2% in the x-direction. This was after a labor intensive tuning and calibration of the motor and steering was done. The experimental evidence showed that the vehicle would require a feedback system to improve accuracy. This result came to the surface only after much work was completed to tune the open-loop system. Then something as simple as changing batteries lead to inaccurate results. Closing the loop will make the vehicle more accurate at executing the optimal trajectories.

The result of this thesis was not just to parallel park a car. It demonstrated a small step towards an autonomous system that can produce a trajectory and then successfully implement the maneuver.

B. RECOMMENDATION FOR FUTURE WORK

One possible future work should be aimed at furthering improvements to the autonomous vehicle. The next step for the vehicle is to convert the wired system to a wireless one. The X-Bee Pro has been a device that has been proven a reliable for this purpose in the wireless communication world.

After the wireless configuration has been set up, closing the control loop for the vehicle velocity and orientation is the next step. The primary feedback would be from the VICON system. The VICON motion caption could be used to feedback the location of

the vehicle as well as the velocity. The knowledge of the desired position during the parking maneuver could be used as an error and a control law used to drive the error to zero. The VICON system was noisy and the feedback signal will therefore have a slight delay, as the last 10 data points would be averaged. The feedback will need to be conducted on the computer sending the commands to the vehicle. Therefore the VICON system should be networked with the computer sending the commands to the vehicle. This action will minimize any additional delays already added to a wireless communications to the vehicle.

Once the above steps are complete, a logical next step is to transfer the optimal control algorithm to the vehicle itself. This might be done with the ArduPilot, but since the computational power of the ArduPilot is limited, a more capable piece of hardware may be required. Completing this step would allow for the vehicle to be fully autonomous to plan and implement trajectories with an onboard feedback system to close the loop. Finally, designing new experiments with moving obstacles, and followed by field-testing.

APPENDIX

Below are the detailed results of the minimum-time parallel parking problem. For every case, there will be first a plot of the Hamiltonian, followed by a plot of the path dual variables, μ , next will be the states and costates and finally the comparison of the optimal solution verses propagation for the inputs of the vehicle's velocity and steering angle. All of the data necessary to implement and validate each solution is presented.

The graph of the path dual variables is a visual confirmation of a situation where the minimum distance between two objects was necessary to complete the time optimal maneuver. There are actually 10 path dual variables in the plots, but they were plotted in the same graph to allow a visual inspection for cases where some of them are not zero. If a minimum distance was reached for a path constraint then, the value for the corresponding μ will be a negative value.

The Hamiltonian and costate values are affected when any μ goes negative. The Hamiltonian will ideally be a constant equal to -1 for minimum-time solutions. If any μ goes negative at any point during the solution, the Hamiltonian will not be exactly negative one.

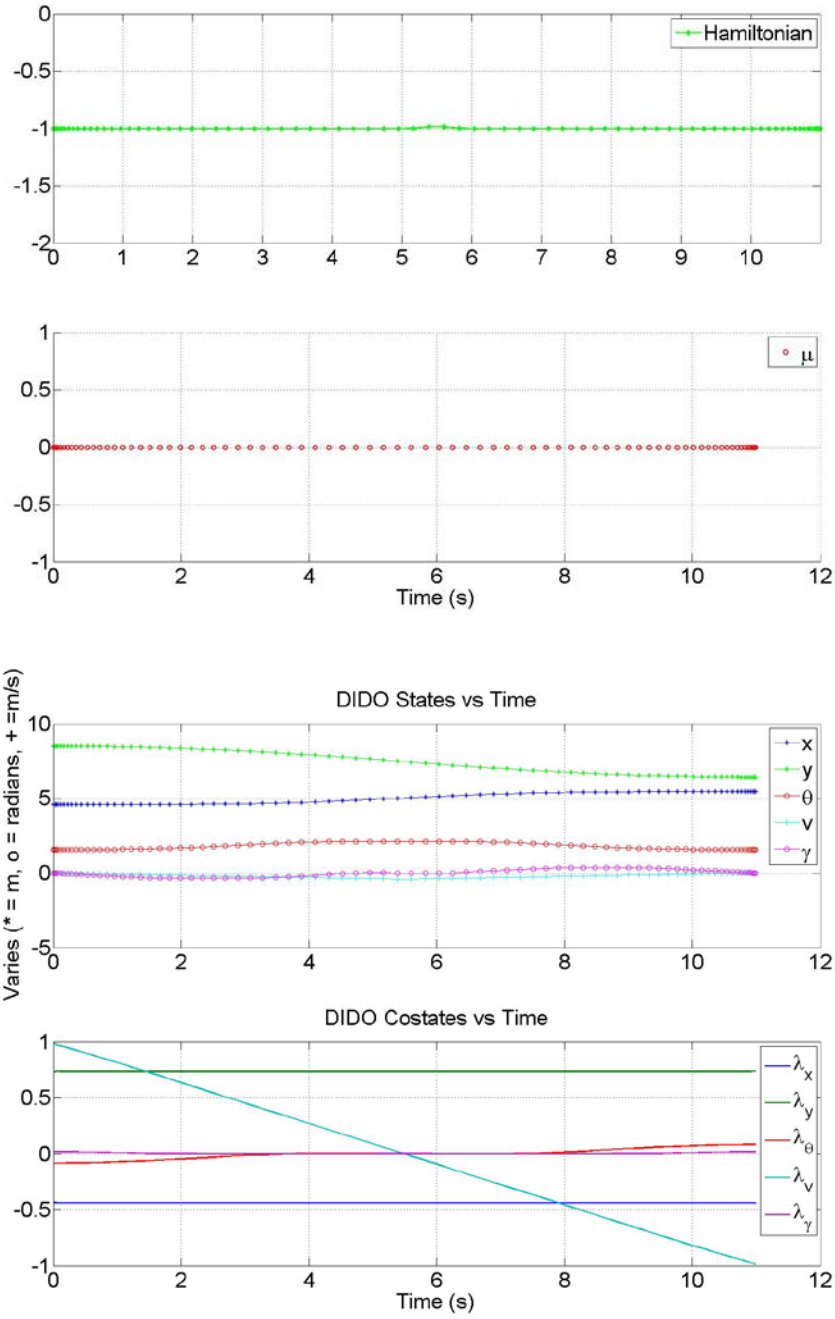
The values of the costates are affected as well, but only at the point when the μ is not zero. The λ_x and λ_y are a constant for an optimal solution. Because of the effects of the path constraint, the λ_x and λ_y shift to another value and then remain constant.

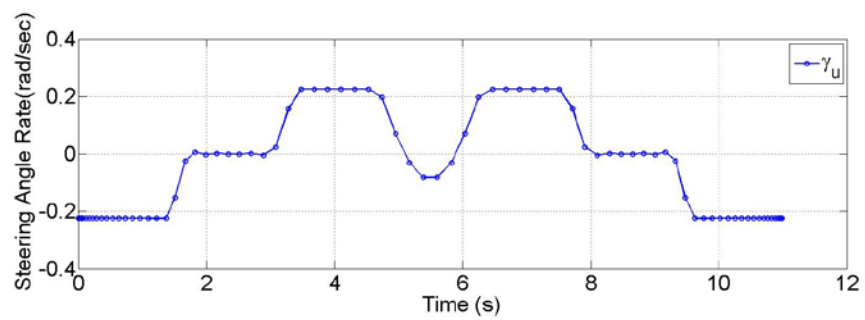
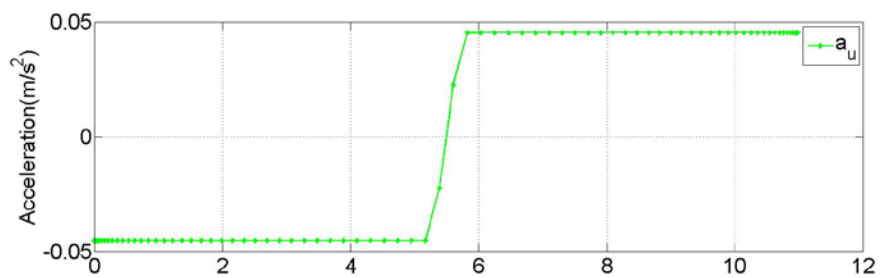
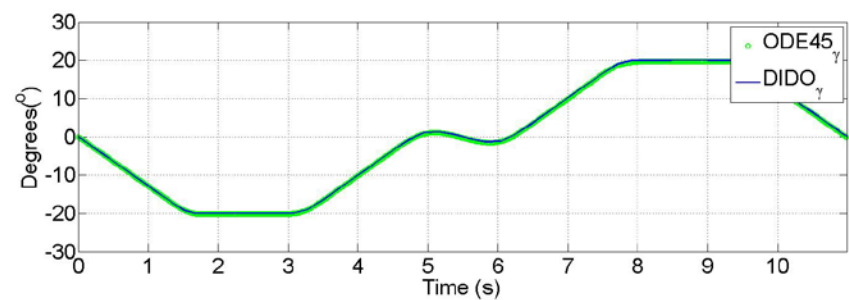
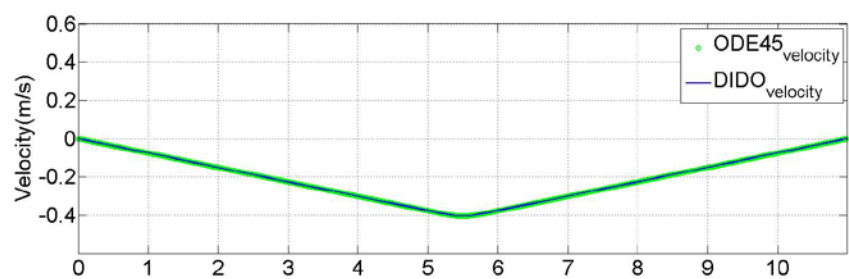
The amount by which nonzero μ influences the Hamiltonian and the co-states depends on how well the path constraint was scaled. The solution of all the scenarios were not scaled for each scenario and case individually, but were scaled only once, to the case that was the most challenging. The most challenging was scenario three, case 3.

To improve the values of the Hamiltonian, each case can be scaled individually. However, doing so made little to no impact on the maneuver trajectories, or their implementation.

Scenario One

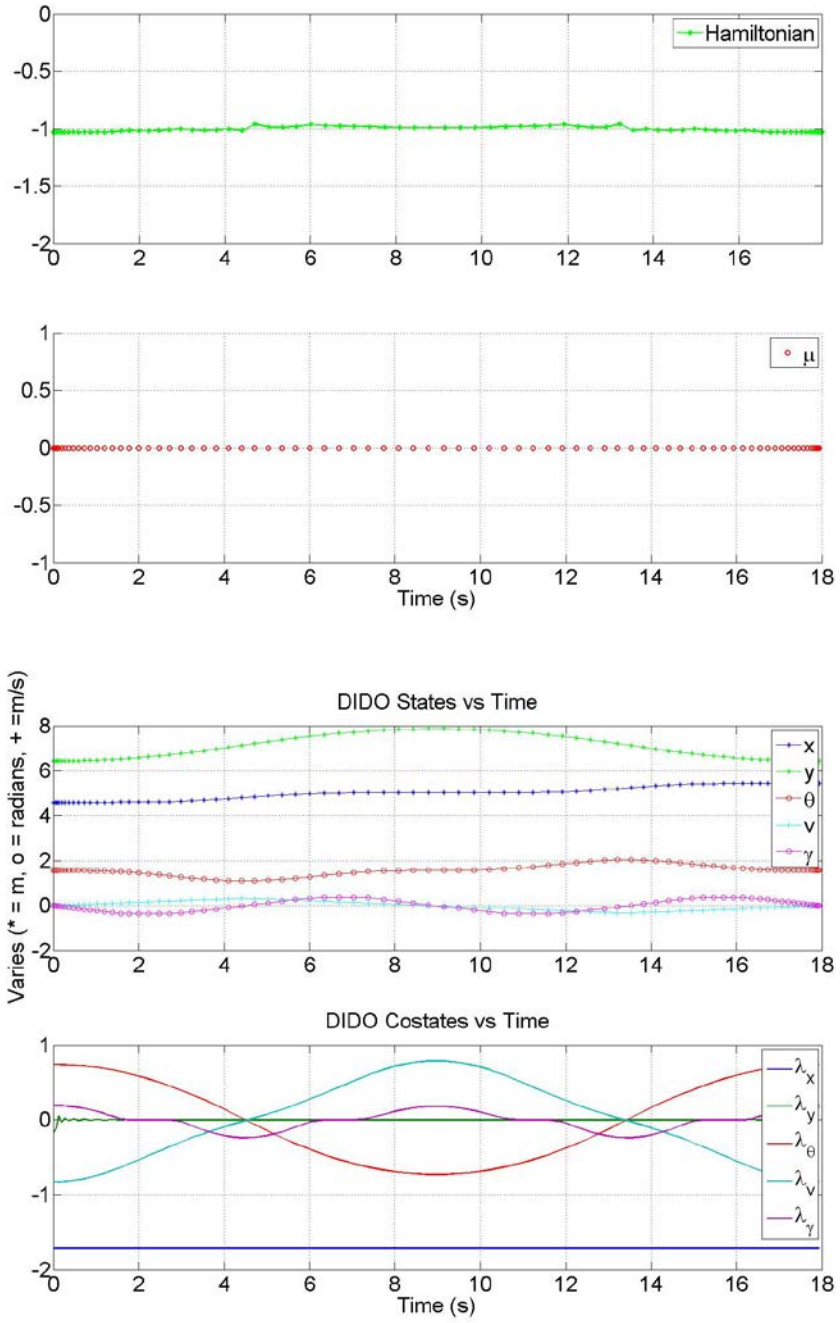
Case 1

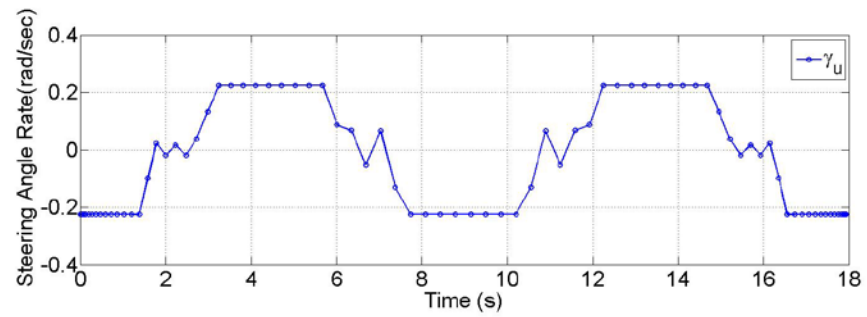
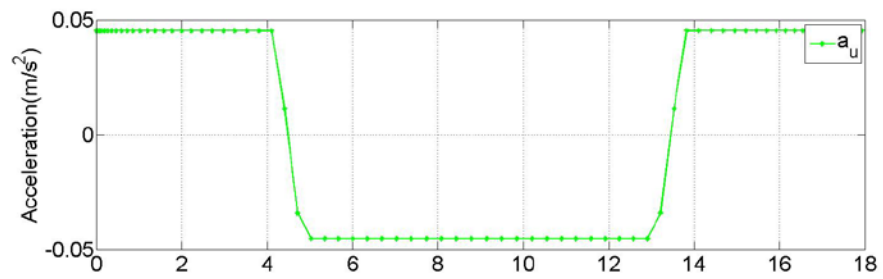
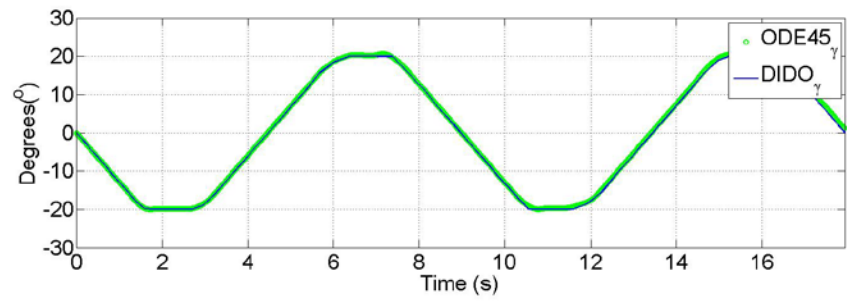
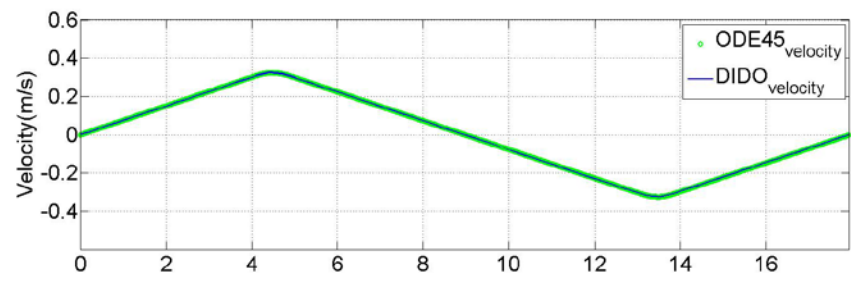




Scenario One

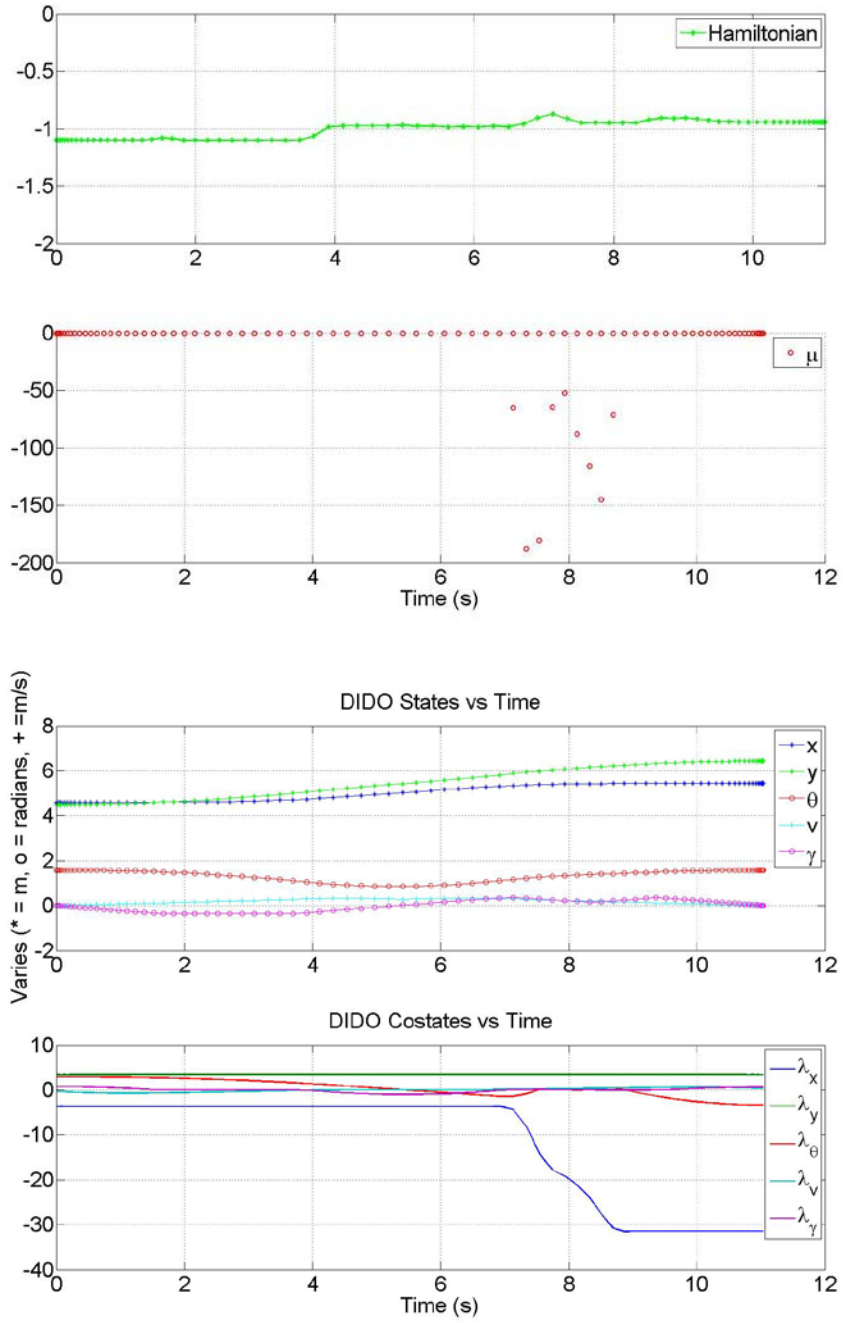
Case 2

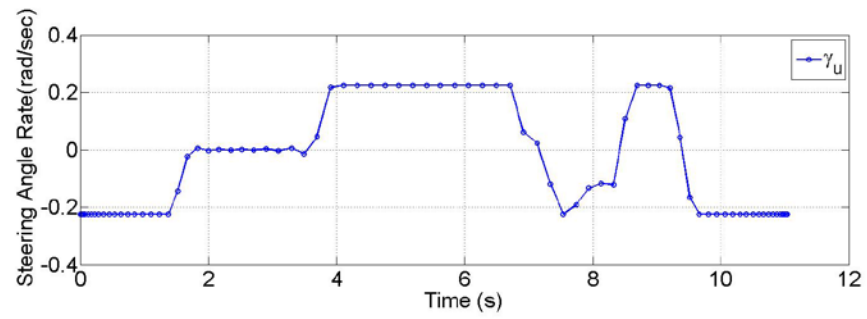
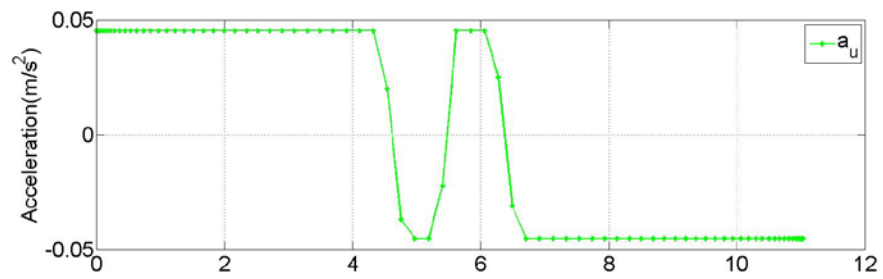
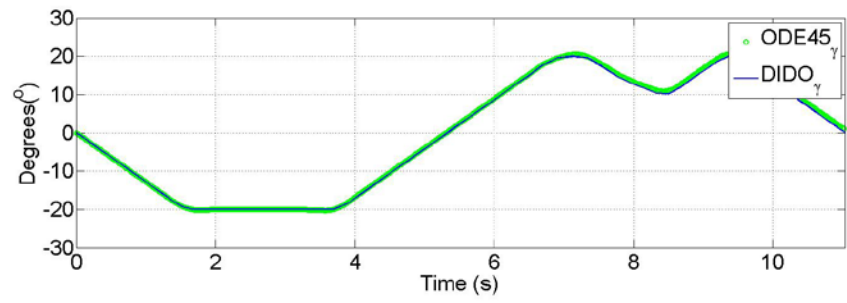
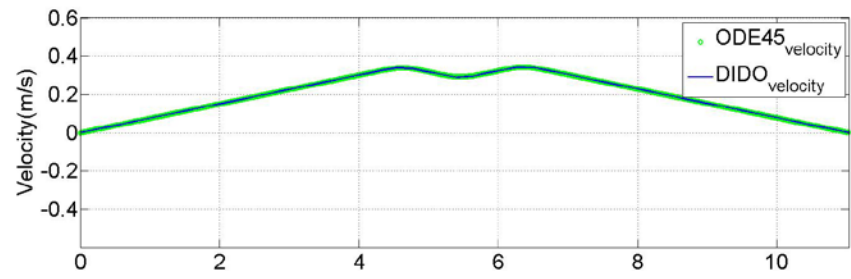




Scenario One

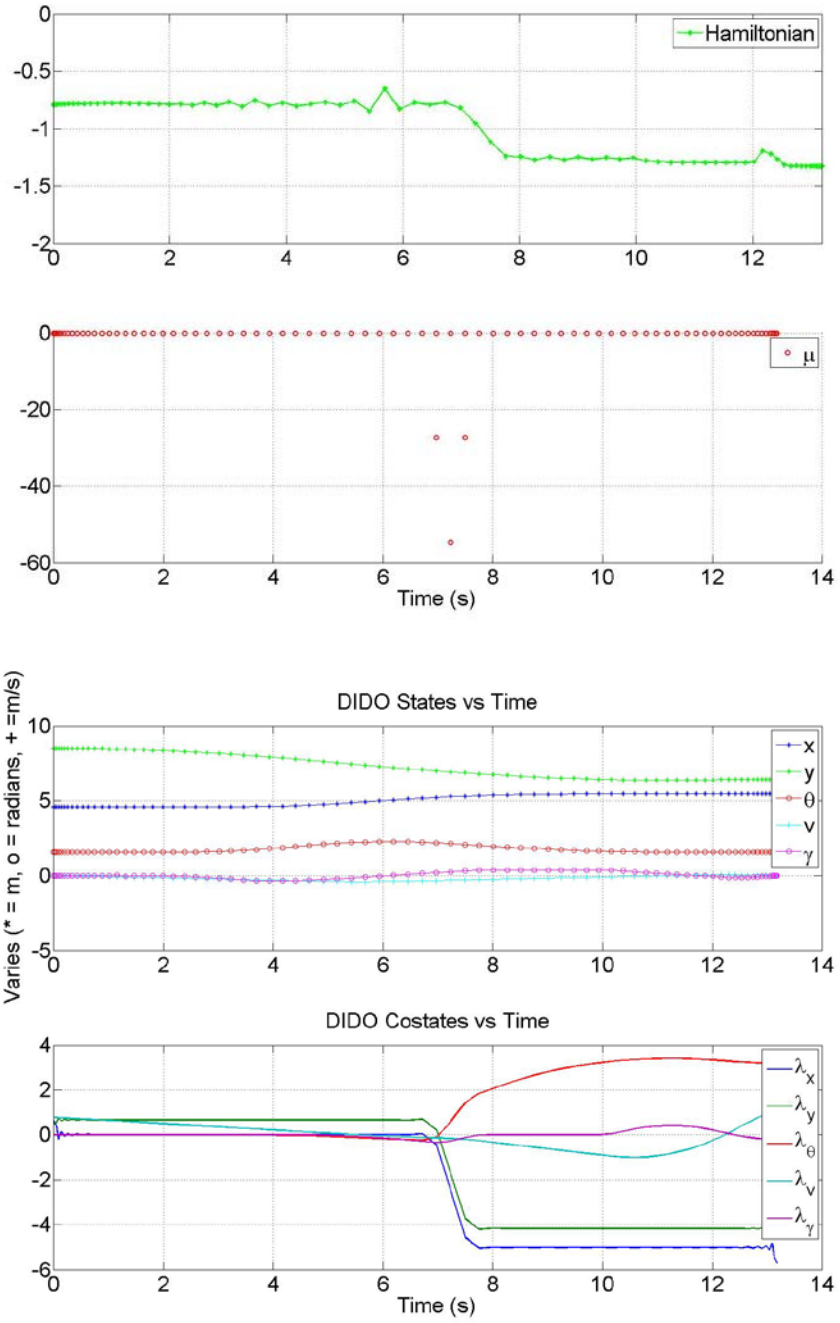
Case 3

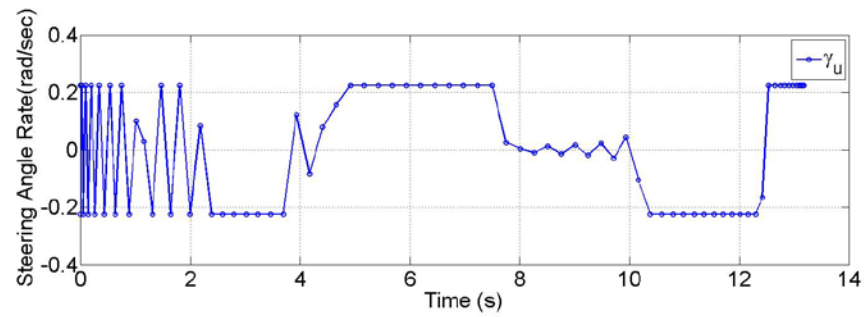
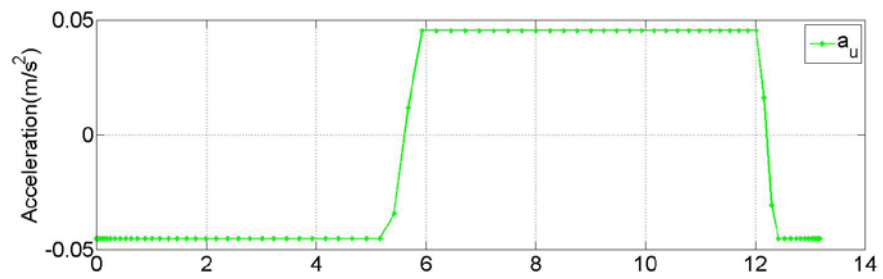
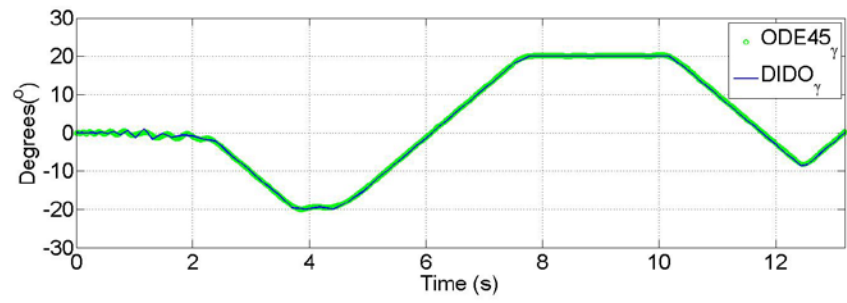
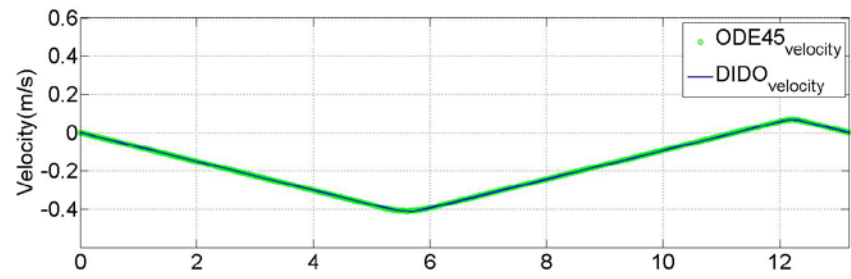




Scenario Two

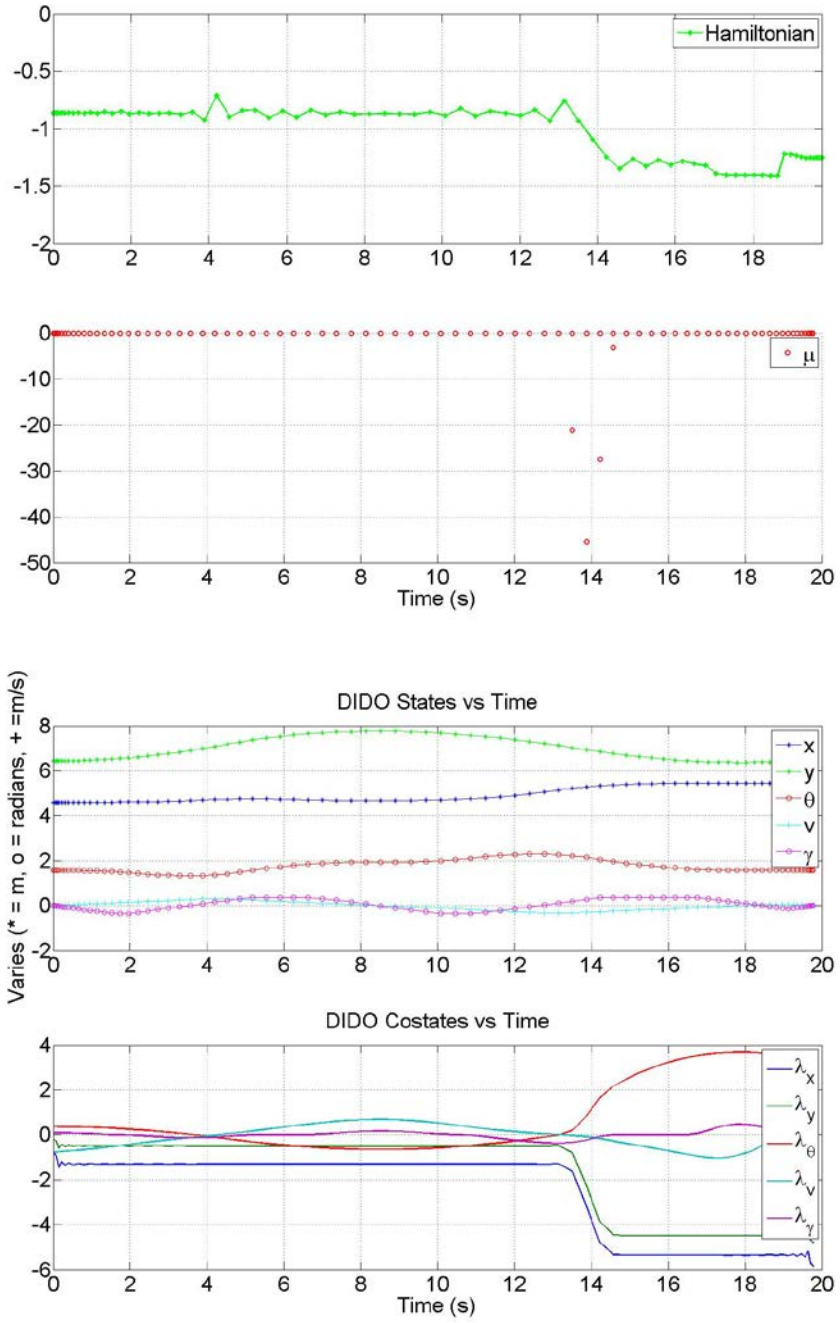
Case 1

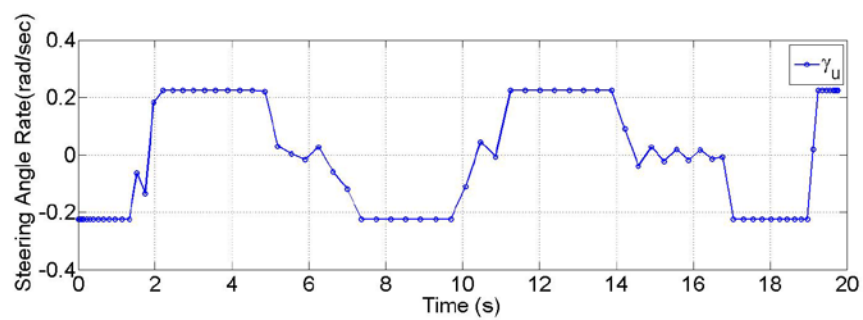
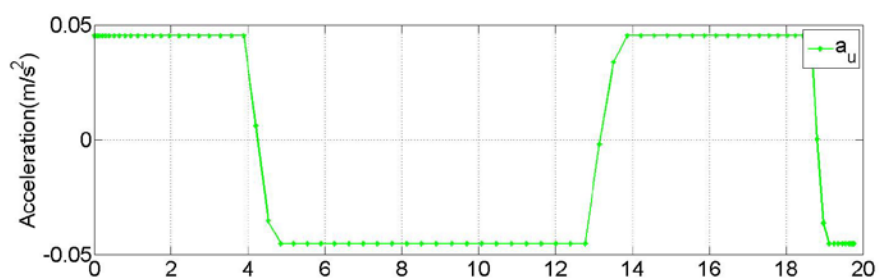
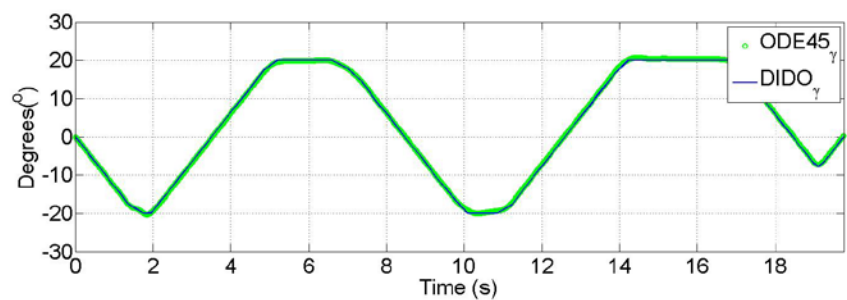
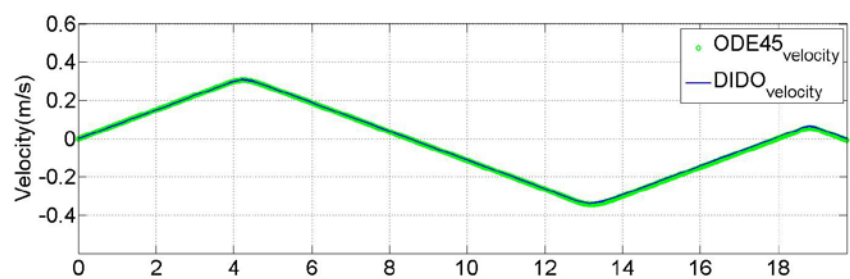




Scenario Two

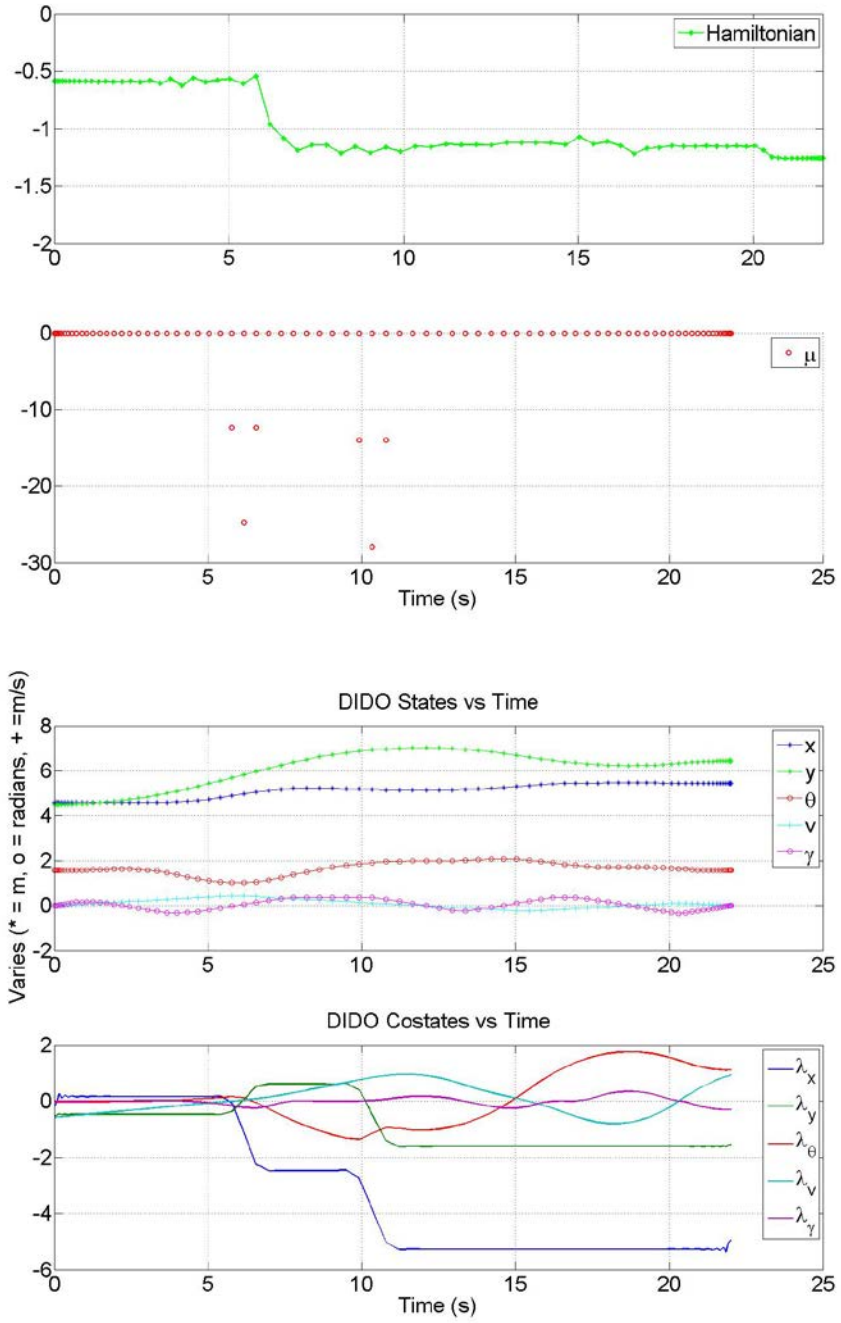
Case 2

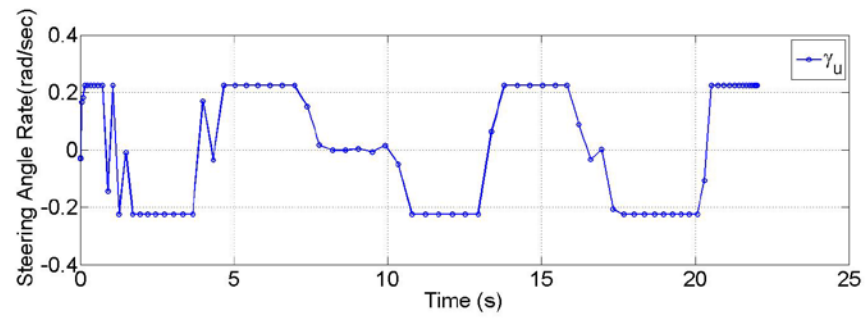
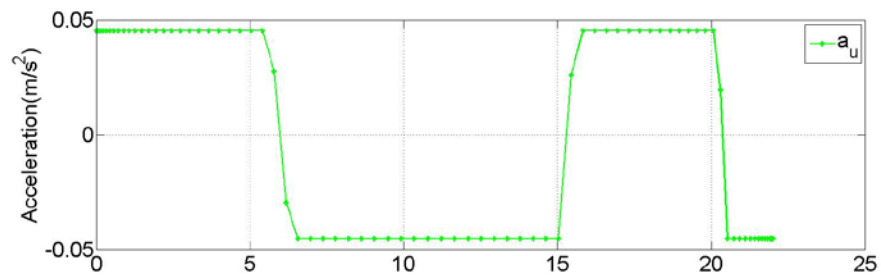
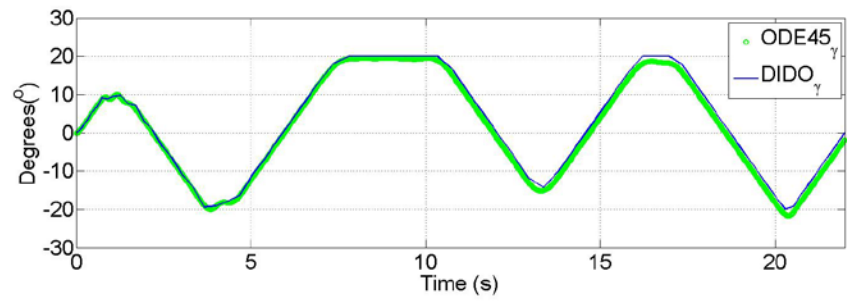
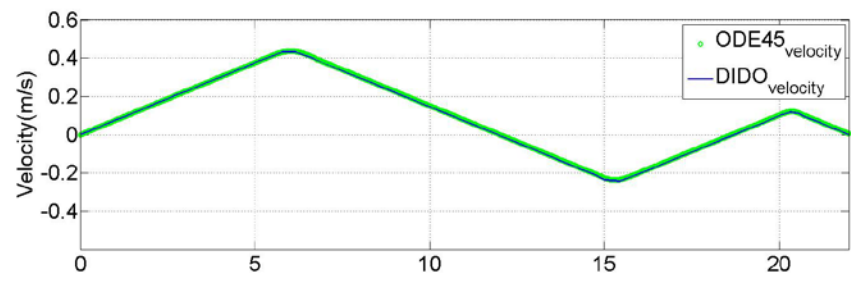




Scenario Two

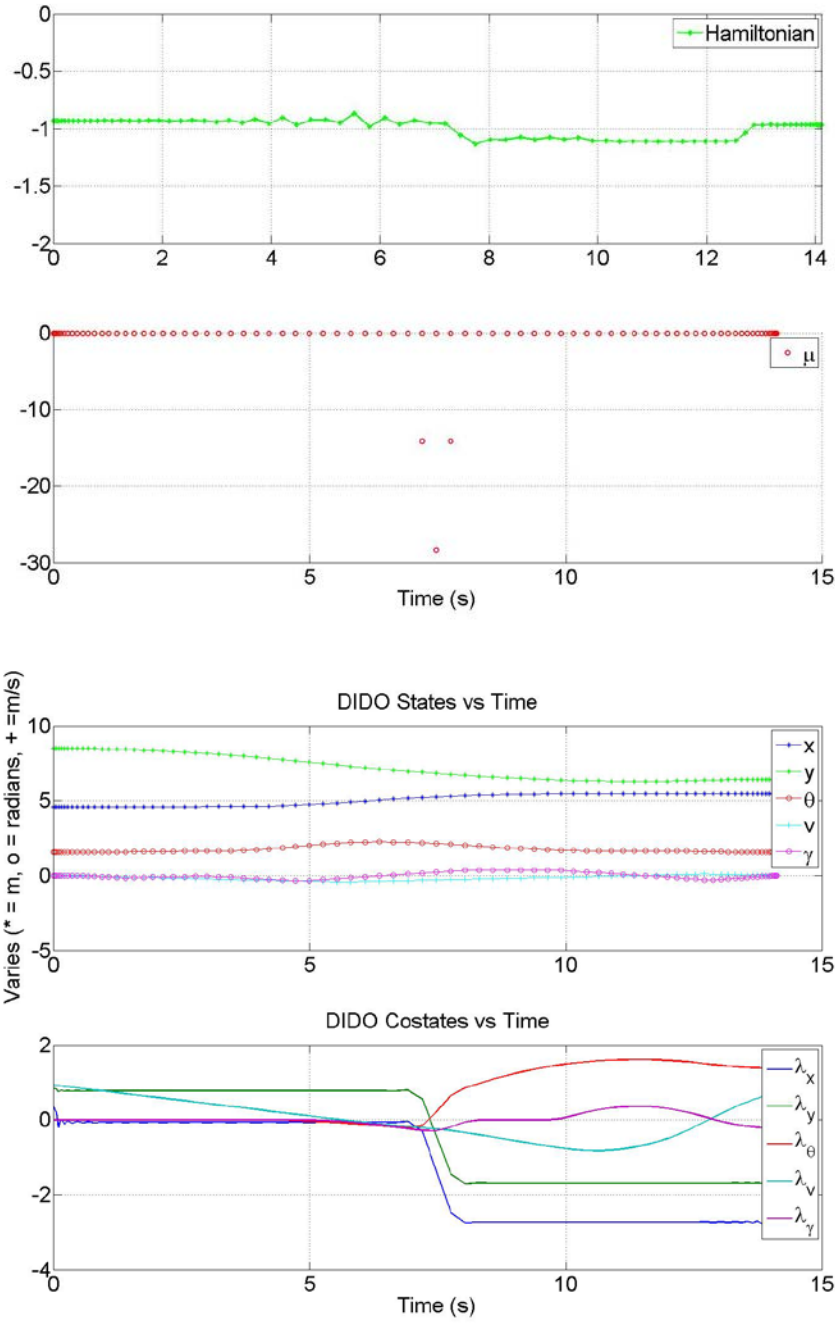
Case 3

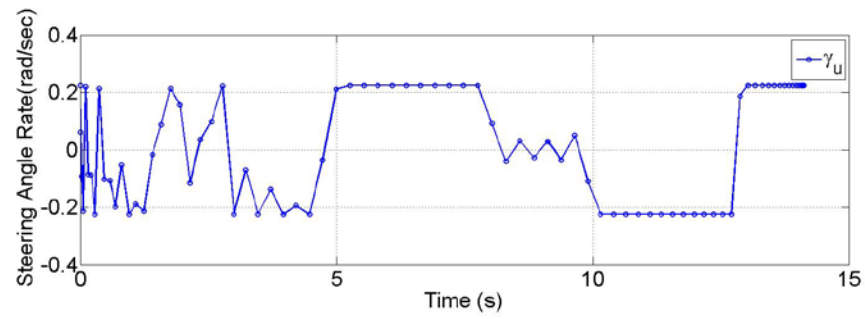
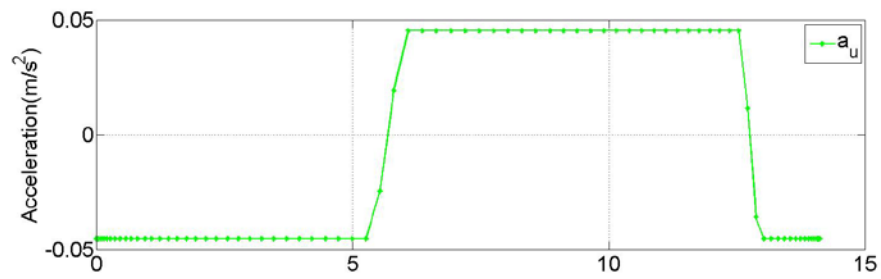
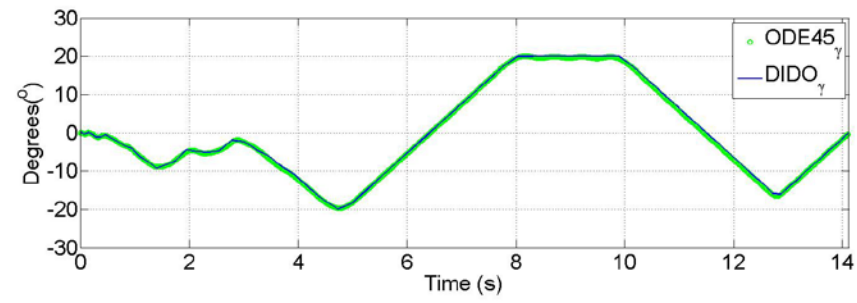
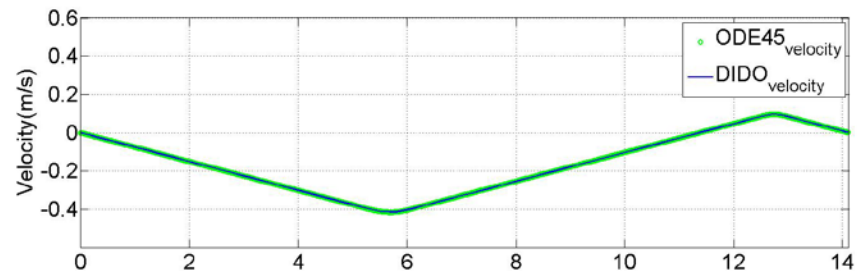




Scenario Three

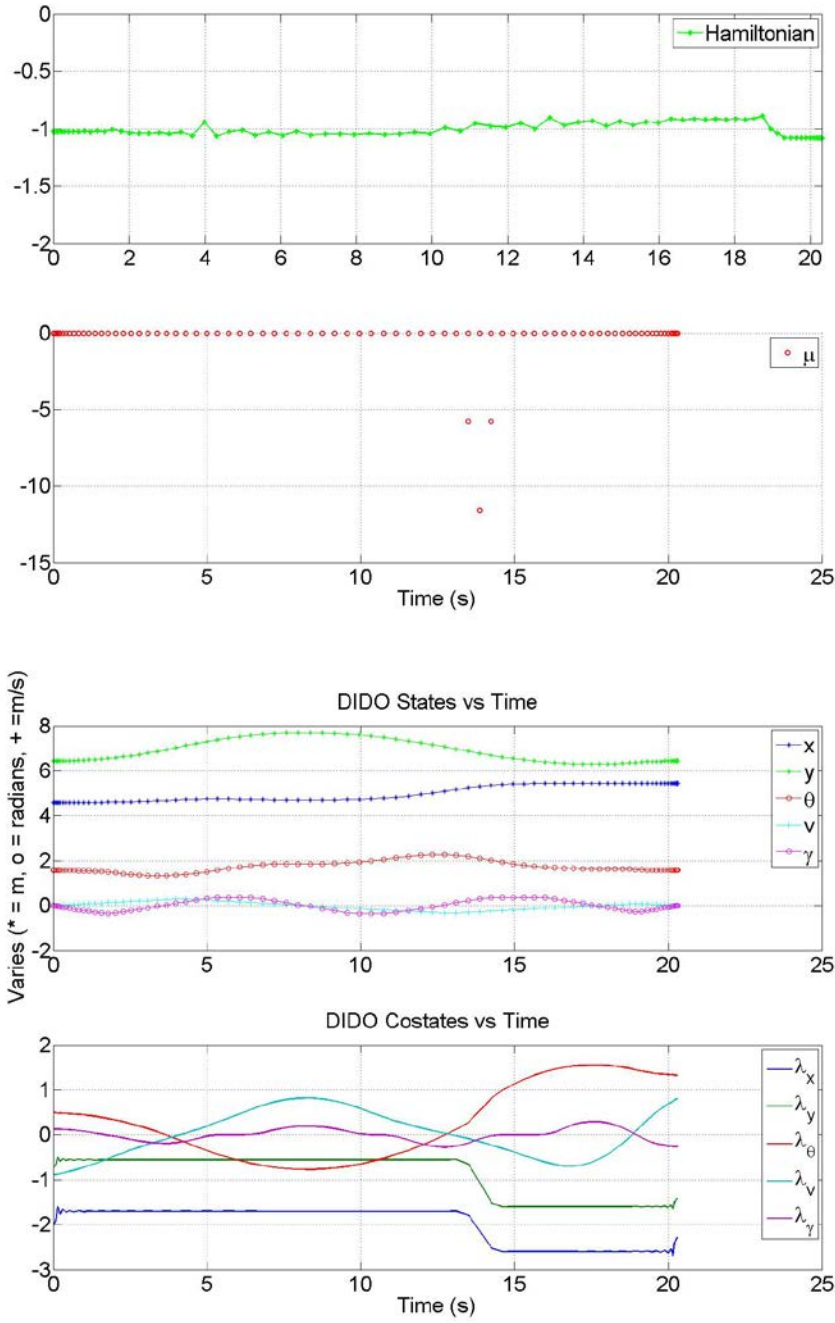
Case 1

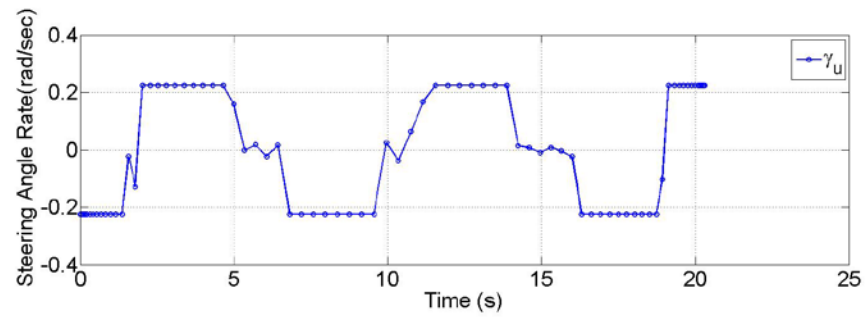
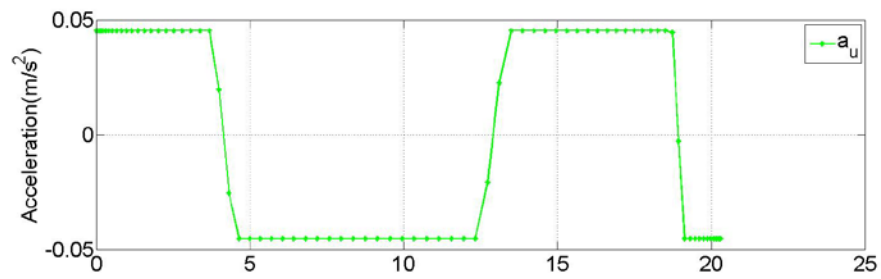
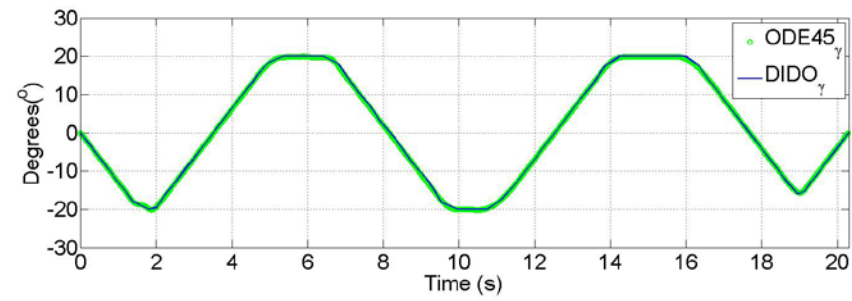
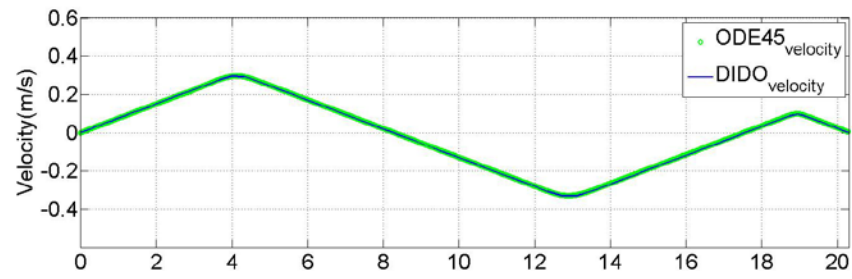




Scenario Three

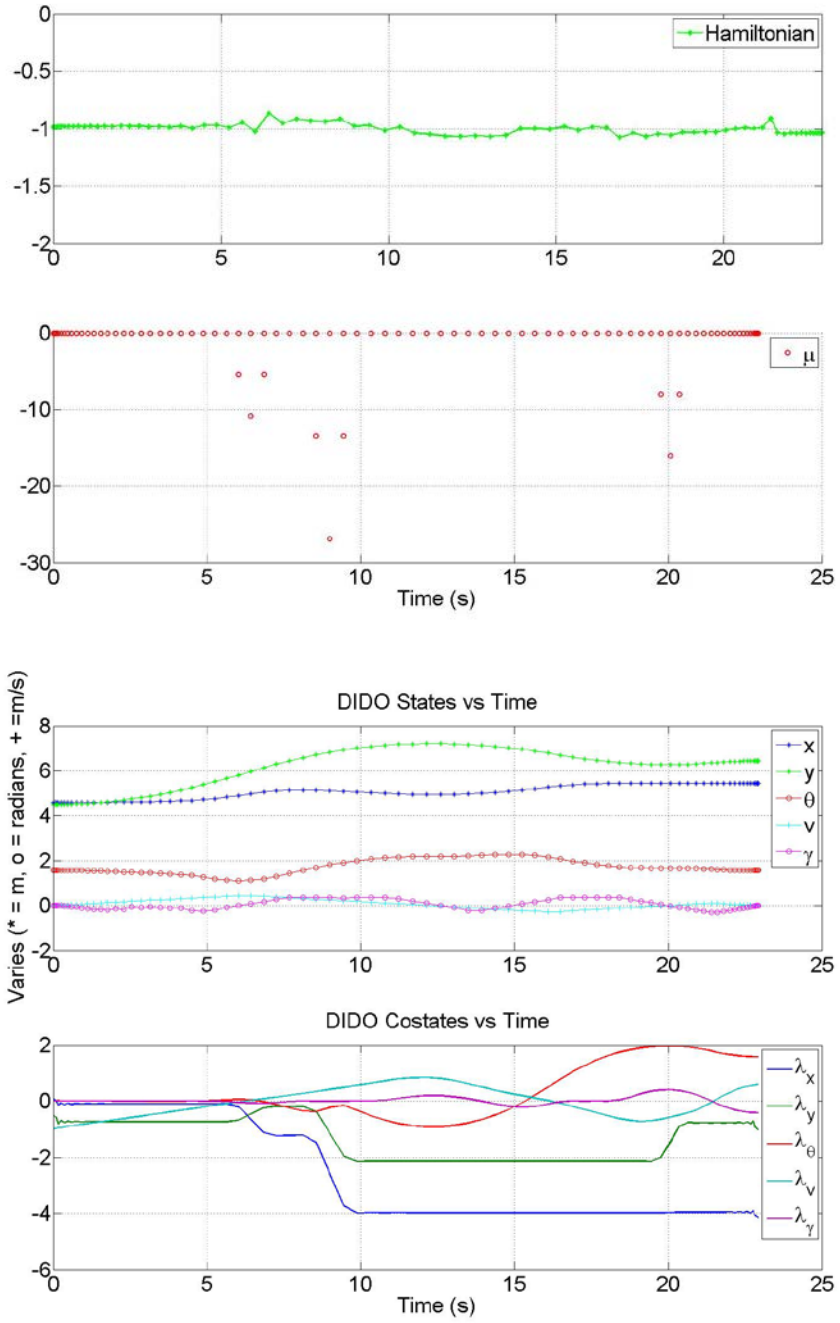
Case 2





Scenario Three

Case 3



THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- [1] K. L. Wagstaff, "Smart Robots on Mars: Deciding Where to Go and What to See." *Juniata Voices*, 2009, pp. 5-13, Sep. 16, 2008.
- [2] M. Morin. "NASA drivers await chance to operate biggest, baddest Mars rover." *LA Times* (Aug. 08 ,2012) Internet: www.latimes.com/news/science, Sep. 21, 2012.
- [3] Jet Propulsion Lab California Institute of Technology. "Curiosity explores Mars." Sep. 29, 2012.
- [4] M. Maimone, J. Biesiadecki, E. Tunstel, Y. Cheng and C. Leger. "Chapter 3: Surface navigation and mobility intelligence on the mars exploration rovers." *Intelligence for Space Robotics*, San Antonia, TX: TSI Press, 2006, pp. 45–69.
- [5] M. Hurni. "An information-centric approach to autonomous trajectory planning utilizing optimal techniques." PhD Dissertation, *Naval Postgraduate School, USA*, 2009.
- [6] I. M. Ross. "A Beginner's guide to DIDO: A MATLAB application package for solving optimal control problems." *Monterey, CA: Elissar Global*, 2007.
- [7] TRAXXAS, "Summit 1/10 scale 4WD Extreme Terrain Monster Truck." Internet: www.traxxas.com/products/models/othermodels/5607summit2010, [Oct. 6, 2012].
- [8] ArduPilot. "Programming the Atmega328p (PPM encoder/failsafe)." Internet: www.code.google.com/p/ardupilot-mega/wiki/AVRStudio, Oct. 22, 2012 [Nov. 13, 2011].
- [9] Spark Fun. "ArduPilot Mega - Arduino Mega compatible UAV Controller." Internet: www.sparkfun.com/products/10294, [Oct. 22, 2012].
- [10] Parallax. "HB-25 Motor Controller." Internet: www.Parallax.com, [Oct. 12 ,2012].
- [11] VICON. "VICON products." Internet: www.vicon.com/products, [Oct. 22 ,2012].
- [12] Department of Motor Vehicles. "California Driver Handbook Parking." Internet: www.dmv.ca.gov, [Oct. 12 ,2012].
- [13] S. R. Blackburn. "The Geometry of Perfect Parking." Internet: http://personal.rhul.ac.uk/uah/058/perfect_parking.pdf, Jul. 18, 2012 [Nov. 30, 2009].

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Mark Karpenko
Naval Postgraduate School
Monterey, California
4. I.M. Ross
Naval Postgraduate School
Monterey, California